UDC 004.8: 621.39 **doi: 10.32620/aktt.2025.3.08**

Viktor MAKARICHEV, Vladimir LUKIN, Sergii KRYVENKO, Iryna BRYSINA

National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine

DIGITAL IMAGE REPRESENTATION BY ATOMIC FUNCTIONS: FEATURES FOR COMPUTER VISION AND MACHINE LEARNING

Digital images obtained from remote sensing (RS) systems have become essential in numerous technological applications across diverse domains, including environmental monitoring, agriculture, urban planning, and defense. These images are typically characterized by high spatial and spectral resolution, resulting in large data volumes. Compared to other data types, their substantial size presents challenges in terms of the efficient application of machine learning (ML) and computer vision (CV) methods. In particular, the processing of such large-scale data can be computationally intensive and time-consuming, making it difficult to deploy conventional ML and CV techniques in scenarios requiring real-time responses or in systems with limited processing resources, such as autonomous platforms. One of the key issues in this context is the development of compact image representations that retain essential features for further analysis. These representations must reduce data dimensionality without losing critical information required for classification, clustering, and other ML/CV tasks. In this study, we explore the discrete atomic transform (DAT), which is based on atomic functions, as a potential solution to this problem. Previous research has demonstrated that DAT provides valuable benefits in terms of data compression and encryption, thereby enabling secure and efficient storage and transmission. The focus of this work is to assess whether DAT is suitable for ML and CV applications, particularly in the context of image clustering. We evaluated the performance of the well-known k-means clustering algorithm when applied to DAT images. The experimental results demonstrate that using DAT significantly reduces computation time, achieving multiple-fold acceleration, without compromising clustering quality. This suggests that DAT not only minimizes data size and preserves structural and statistical features relevant to learning-based tasks. These results indicate that the integration of DAT into preprocessing pipelines for RS imagery is a promising approach. The proposed method can enhance the efficiency of downstream ML and CV algorithms, especially in constrained environments where computational resources are limited. Overall, the discrete atomic transform is a practical and versatile method for improving the scalability and applicability of intelligent image analysis in remote sensing and related fields.

Keywords: *image representation; atomic function; discrete atomic transform; atomic embeddings; image clustering.*

1. Introduction

1.1. Motivation

Remote sensing (RS) imagery has become a key data source for diverse fields including ecological monitoring [1], precision agriculture [2, 3], urban development [4], disaster response [5, 6], oceanographic studies [2, 7], and defense [8]. Modern RS sensors are capable of capturing extremely high-resolution images –often with tens of millions of pixels per frame. While this improves analytical potential, it simultaneously introduces significant challenges related to memory usage, transmission, and computational load [9]. Standard image compression techniques [10] help mitigate some of these limitations [11], but when RS images are subjected to machine learning (ML) or computer vision (CV) algorithms, new bottlenecks emerge.

ML and CV methods frequently operate on raw or decompressed images [12,13], demanding substantial memory and processing time. Solutions such as GPUs, TPUs, and distributed computing have been adopted to reduce latency [14], but their deployment is impractical in many real-world systems [15]. For example, edge devices and autonomous platforms are typically constrained by limited power, compute capacity [16], and network bandwidth [17]. Moreover, transferring large RS images via communication links can be infeasible due to bandwidth limitations or interference. This creates an urgent need for compact image representations that reduce data size while preserving relevant features for ML/CV analysis. Conventional compression algorithms were not designed with learning tasks in mind. As such, the integration of compact representations that combine efficient compression with learning-readiness becomes a compelling research direction. Prior studies of the discrete atomic transform (DAT) have demonstrated its strong

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

compression and approximation capabilities, while more recent work shows that lossy DAT compression introduces only minor accuracy loss for classification tasks. These results suggest that DAT might enable fast and efficient learning, especially in unsupervised scenarios. The potential of DAT to accelerate key algorithms such as clustering without degrading their output quality represents a valuable opportunity for further exploration.

1.2. State-of-the-art

The exponential growth in the use of remote sensing (RS) imagery has led to significant efforts aimed at reducing the computational burden associated with processing such large-scale data. High-resolution images generated by modern RS systems are routinely used in ecological, agricultural, urban, and security applications, but their vast size creates constraints on memory, transmission, and real-time processing. Compression algorithms have been widely adopted to address storage and bandwidth limitations. However, these solutions alone do not resolve challenges related to data analysis using machine learning (ML) and computer vision (CV) tools. ML and CV techniques extract meaningful patterns from data and often work with full-resolution or decompressed imagery, making the analysis process computationally expensive. To alleviate this, solutions involving GPU acceleration, cloud computing, and distributed systems have been implemented. Yet, such architectures are not always feasible for real-world edge devices and autonomous systems due to constraints in power and connectivity.

While many general-purpose and image-specific compression methods exist [18], they are traditionally optimized for human visual perception and efficient storage rather than ML/CV compatibility [19]. As a result, they may discard features that are semantically meaningful for automated analysis [20]. There is growing interest in developing representations that are both compact and learning-aware.

The discrete atomic transform (DAT), which is based on atomic functions, has shown promise in this direction [21]. Recent studies demonstrate that DAT achieves good compression ratios while preserving important image characteristics [22]. Moreover, DAT has been shown to maintain classification accuracy under lossy compression [23], indicating its potential utility in broader ML/CV applications such as image clustering [24] and segmentation [25].

1.3. Objectives and the approach

This research aims to evaluate whether image representations based on the discrete atomic transform (DAT) can serve as effective input for machine learning (ML) and computer vision (CV) tasks, particularly under resource-constrained conditions. Traditional compression algorithms are not optimized for learning-based applications, leading to inefficiencies when decompressing data for ML/CV analysis.

Our objective is to assess the DAT's ability to reduce image size while maintaining clustering performance in unsupervised learning settings. We focus on the widely used k-means algorithm as a benchmark task. To this end, we propose and test two modifications of the standard k-means algorithm that operate directly on DAT-based image representations. The aim is to achieve similar clustering quality with significantly lower computational costs. This work aligns with current efforts to develop edge-friendly ML strategies by providing a scalable, efficient alternative to full-resolution image processing in RS applications.

2. Materials and methods of research

Denote by M a matrix of a given h×w digital image I. In this research, we consider three-channel RS images. However, the proposed approaches can be generalized for other cases.

Let T be an image transform that maps M to its representation W:

$$T: M \rightarrow W$$

To train a neural network, it is important to choose a dataset that clearly and accurately labels objects. Also, the dataset should allow training using different image sizes, i.e., it should be of sufficiently high resolution. These parameters are important when choosing a dataset [21]. By the suggested definition, W is an ML/CV-oriented representation of I with respect to a given algorithm A if the following conditions are satisfied:

- Storing W requires less memory than storing M. In other words, data compression is provided.
- W can be represented as a set of matrix blocks $B = \{B_1, B_2, \ldots, B_N\}$, such that $A(M) \approx A(P)$, where $P = \{B_{i1}, B_{i2}, \ldots\}$ is a subset of B ($L < N$), and "≈" denotes a minor deviation of an output of the algorithm A in terms of an appropriate metric(s). This feature means that a given ML/CV method can be applied to smaller data without significant degradation of its efficiency, and, hence, computations could be reduced.

In our previous research [24, 26], we investigated the lossy and lossless compression features of image representation by AFs. Based on them, discrete atomic compression (DAC) was developed. As shown, DAC provides considerable memory savings, especially when compressing satellite data [26]. Thus, the first requirement is met. Further, to provide the correctness of the

comparison A(M) ≈ A(P), one should specify P, how to apply A to P, and how to measure the difference. In this research, we consider the k-means clustering algorithm and image representation W provided by AFs. Therefore, the main tasks are as follows:

- Indicate which part of the suggested representation could be used in the k-means algorithm to cluster a given image without significant modifications that might improve computations;
- Provide an appropriate update of the k-means algorithm;
- Specify metrics for measuring efficiency;
- Perform the comparison of the designed modifications in terms of the selected indicators.

We start with the properties of atomic functions and their applications to image processing.

3. Image Representation by Atomic Functions

3.1. Atomic Functions and their Properties

In function theory, a function is called atomic if it is a solution with a compact support of a linear functional-differential equation with constant coefficients and linear transformations of an argument [22]. This class of functions can be considered a generalization of wavelets [27].

Here, we consider a set of atomic functions given as follows:

$$\mathrm{up_s}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{itx} \prod_{k=1}^{\infty} \frac{\sin^2(st(2s)^{-k})}{s^2 t(2s)^{-k}\sin(t(2s)^{-k})}\,dt, \quad (1)$$

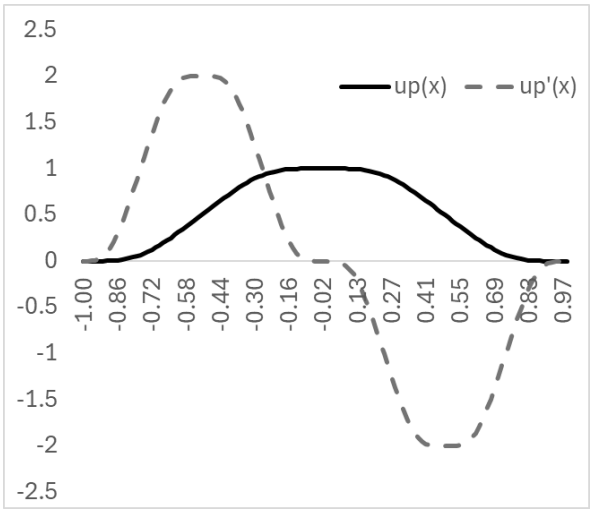
where s = 1, 2, 3, ….

A support of $\mathrm{up_s}(x)$ is a segment [-1, 1], i.e. $\mathrm{up_s}(x) = 0$ if $|x| \geq 1$. Also, for any s, the function $\mathrm{up_s}(x)$ satisfies the equation

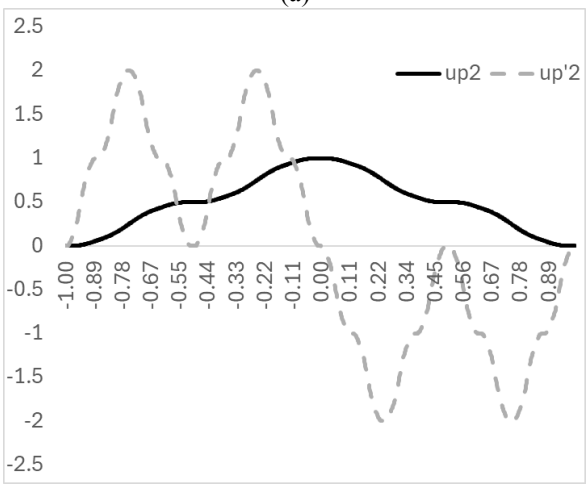
$$y'(x) = 2\sum_{k=1}^{2s} (y(2sx + 2s - 2k + 1) - y(2sx - 2k + 1)). \quad (2)$$

The equation (2) has a simple interpretation. The graph of the derivative of $\mathrm{up_s}(x)$ consists of several parts, each of which is similar to the graph of $\mathrm{up_s}(x)$. Figure 1 illustrates this property.

Despite the complexity of the representation (1), atomic functions $\mathrm{up_s}(x)$ are simple to apply. First, their values can be found precisely in points of a dense grid using fast algorithms that leverage the principles of dynamic programming [28]. Next, their derivatives of any order can be easily computed using (2). Finally, due to the compactness of the support of $\mathrm{up_s}(x)$, various numerical expansion schemes, which are based on these mappings, have low time and spatial complexity [29].



(a)



(b)

Fig. 1. Graphs of atomic functions and their derivatives: (**a**) $\mathrm{up}_1(x)$ and (**b**) $\mathrm{up}_2(x)$

There are several ways to apply atomic functions as constructive tools. One of them is the use of linear combinations of translates of $\mathrm{up_s}(x)$:

$$\varphi(x) = \sum_k c_k \mathrm{up_s}\left(x - \frac{k}{N}\right). \quad (3)$$

Spaces of functions of the form (3) were investigated in [22, 23]. It was shown that these spaces contained algebraic polynomials. Moreover, in their periodic versions, there exists a basis that consists of functions close to trigonometric. In addition, such spaces are asymptotically extremal for approximating differentiable functions. Hence, atomic functions $\mathrm{up_s}(x)$ have good approximation properties that guarantee insignificant errors when expanding functions to the form (3). A combination of these features makes $\mathrm{up_s}(x)$ as good as trigonometric functions for signal and image processing [20, 27]. Further, we consider the practical aspects of their applications.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

3.2. Discrete Atomic Transform

When applying a certain system of functions to processing and analyzing discrete data, an ability to get orthogonal components is preferable [27]. Expansion of a function to the sum of orthogonal terms is one of the core principles of applying atomic functions to image processing [29]. Consider it in more detail.

Denote by L_0 the linear space of functions of the form (3). It follows that, in this space, there exists a set of nested subspaces $\{L_k\}$, such that

- $L_0 \supset L_1 \supset L_2 \supset \ldots$;
- any function $f \in L_k$ can be given in the form

$$f(x) = \sum_j c_j v_k\left(x - 2^{k+1} \cdot \frac{j}{N}\right),$$

where $v_k(x)$ is an infinitely smooth function with compact support:

$$v_k(x) = 0, |x| \geq 2^{k+1}/N. \qquad (4)$$

These conditions mean that each subspace L_k consists of linear combinations of shifts of a single function $v_k(x)$ with compact support. Moreover, by (4), each basic function $v_k(x)$ is two times less local than $v_{k-1}(x)$. In other words, the spaces $\{L_k\}$ provide different levels of resolution, differing by a factor of two.

Further, let W_k be the orthogonal complement to L_k in the space L_{k-1} with respect to the inner product defined by the formula: $(f, g) = \int_{-\infty}^{\infty} f(x)g(x)dx$.

This implies that, for any positive integer m, the following orthogonal decomposition holds:

$$L_0 = W_1 \oplus W_2 \oplus \ldots \oplus W_m \oplus L_m.$$

Besides, in each space W_k, there is a basis $\left\{w_k\left(x - 2^{k+1} \cdot \frac{j}{N}\right)\right\}_{j=0,\pm1,\pm2,\ldots}$ that consists of shifts of the function $w_k(x)$ with a support $[0, 2^k \cdot 3/N]$.

The system of functions

$$\left\{w_k\left(x - 2^{k+1} \cdot \frac{j}{N}\right), v_m\left(x - 2^{m+1} \cdot \frac{j}{N}\right)\right\}_{k,j} \qquad (5)$$

constitutes a basis of the space L_0. Each function $f \in L_0$ has the following expansion:

$$f(x) = \sum_{k=1}^{m} \sum_j \omega_j^{[k]} w_k\left(x - 2^{k+1} \cdot \frac{j}{N}\right) + \sum_j \omega_j^{[m+1]} v_m\left(x - 2^{m+1} \cdot \frac{j}{N}\right). \qquad (6)$$

The system (5) is called the atomic wavelet system. Figure 2 shows an example of the expansion (6). Wherein, $\sum_j \omega_j^{[m+1]} v_m\left(x - 2^{m+1} \cdot \frac{j}{N}\right)$ or L_m-components represent the low-frequency profile of a given function f(x). Combining it with W_k-components (for $k = m, m-1, \ldots, 1$, sequentially) provides a reconstruction of the source function f(x). Figure 3 is an illustration.

The concept of the system $\{L_k\}$ is similar to multiresolution analysis (MRA), which is one of the core principles of wavelet theory [27]. Since functions $v_k(x)$ are different, this system belongs to a non-stationary type of MRAs [30].

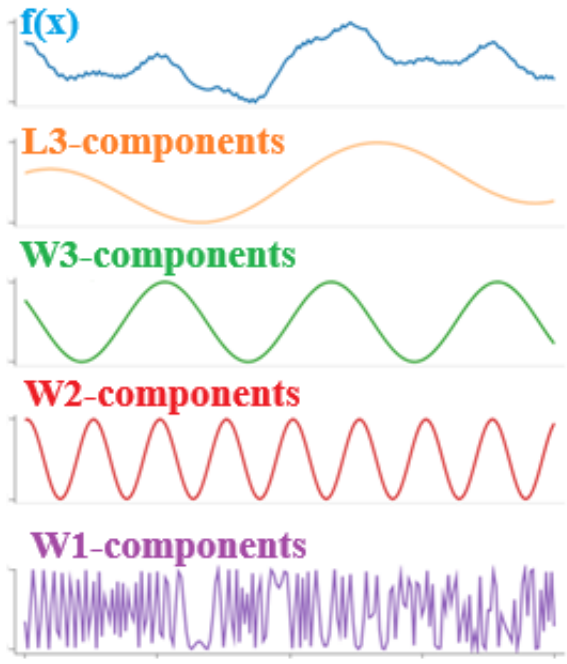


Fig. 2. Sample expansion of a function to the atomic wavelet system (5): graphs of profiles corresponding to different terms of the representation (6)

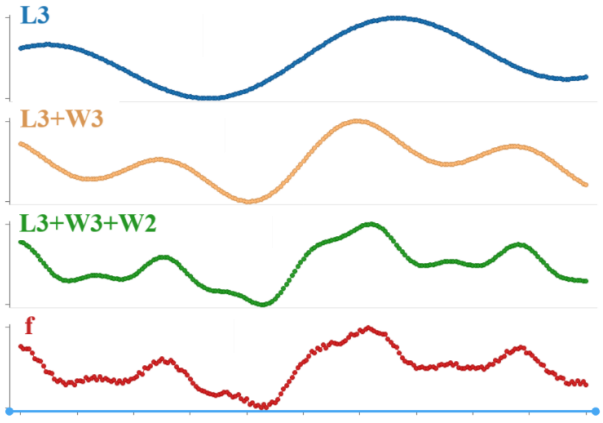


Fig. 3. Sample reconstructions of a given function (red) by items of its atomic wavelet expansion. Adding more terms provides a profile (green) closer to the source

Each function $w_k(x), v_k(x)$ is infinitely smooth and has compact support. So, (5) presents a system of non-

stationary, infinitely differentiable wavelets with a compact support. In practice, an application of wavelet systems constructed using a single function might be preferable due to the simplicity of numerical algorithms and their implementations. However, stationary systems of infinitely smooth, compactly supported wavelets do not exist [30].

The compactness and locality of the support of functions $v_k(x)$ and $w_k(x)$ ensure the design of rapid expansion schemes and the reduction of accumulated errors. The high order of smoothness of these functions makes them useful, especially for representation of smooth signals. Furthermore, they guarantee a good approximation of a wide range of mappings due to the constructive properties of the space L_0 mentioned above.

Let d(x) represents a given one-dimensional vector $D = (d_1, d_2, \dots, d_n)$. Denote by $\Omega = \left(\omega_j^{[k]}\right)$ an ordered vector of its atomic wavelet expansion coefficients (see (6)). A procedure that maps D to Ω is called discrete atomic transform (DAT). Ω can be presented as follows: $\Omega = (\Omega_0, \Omega_1, \dots, \Omega_m)$, where $\Omega_k = \left(\omega_j^{[m+1-k]}\right)$, $k = 0,1,\dots,m$. Here, m is called the depth of DAT. Figure 4 shows an example of DAT of depth 3. Further, we consider two-dimensional DAT and its application to image processing.

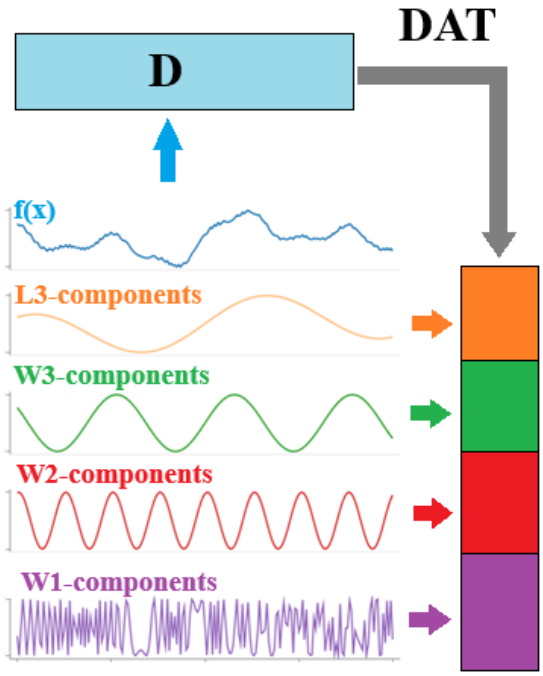


Fig. 4. The structure of a discrete atomic transform of depth 3: the correspondence of DAT coefficients to different frequency bands

3.3. Atomic Embeddings of Images

There are several ways to present digital images. One of the most common is the use of matrices. In this research, we concentrate on this approach. Discrete atomic transform of matrices is constructed using the array transform DAT as follows [26].

Consider two-dimensional matrix M. Apply the array transform DAT of the depth m to each row of this matrix. A matrix B of intermediate DAT-coefficients is computed. Then, apply the array DAT of the depth n to each column of B. A block-structured matrix Ω of DAT-coefficients is obtained. The transform, which maps M to Ω, is called the matrix discrete atomic transform of **the scheme DAT1**. Figure 5 illustrates DAT1. It follows that $\Omega = (\Omega_{ik})_{i,k=0}^{m,n}$, where each block Ω_{ik} contains DAT-coeffcients corresponding to the same frequency band. The number of blocks is equal to $(m+1)(n+1)$.

Consider another scheme of the matrix DAT. Apply the array DAT of depth 1 to each row of a given matrix M and then to each column of the obtained matrix of intermediate DAT-coefficients (see Figure 6). This transform is called **DAT2 of depth 1**. Generally, **DAT2 of depth m** is constructed as follows. DAT2 of depth 1 is performed m times; each time this transform is applied to the block of the lowest frequency band, i.e., to the upper left block (see Figure 7). It is obvious that DAT2 of depth m produces $3m + 1$ blocks.

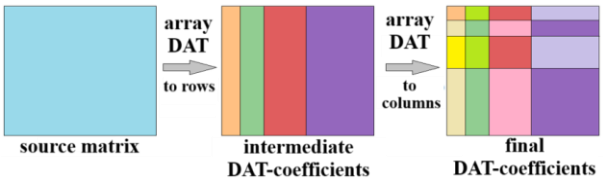


Fig. 5. Discrete atomic transform of matrices: the scheme DAT1. First, the array DAT is applied to each row of a given matrix, and a matrix of intermediate DAT-coefficients is computed. Second, the array DAT is applied to each column of this matrix. The array DAT of depth 3 is used in the given example

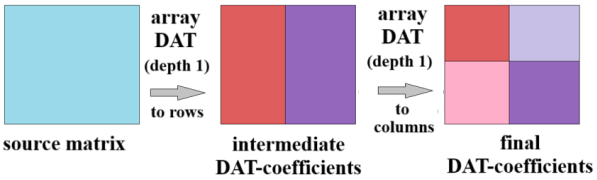


Fig. 6. Discrete atomic transform of matrices: the scheme DAT2 of depth 1. First, the array DAT of depth 1 is applied to each row of a given matrix, and a matrix of intermediate DAT-coefficients is computed. Second, the array DAT of depth 1 is applied to each column of this matrix

Both DAT1 and DAT2 produce a set of blocks. To solve the data compression task, values of these blocks are quantized and then encoded. This method is called discrete atomic compression (DAC) [26].

The DAC algorithm can be applied to both one- and multi-channel images. When compressing full-color 24-

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

bit images given by a matrix M of red, green, and blue components (RGB), the color space transform RGB-to-YCrCb is applied at the first step. It produces three matrices Y, Cr, and Cb of luma and chroma components. Next, DAT is applied to each of them. After that, the computed matrices of DAT-coefficients are quantized and then encoded, usually employing a combination of Golomb coding (GC) and context adaptive binary arithmetic coding (CABAC) [21, 26]. Figure 8 presents the general scheme of color image DAC.

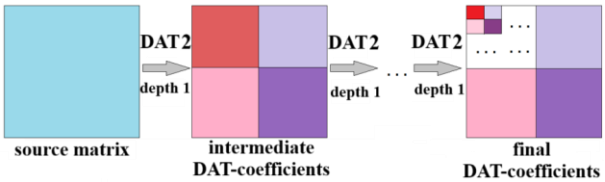


Fig. 7. Discrete atomic transform of matrices: the scheme DAT2, general case. The transform DAT2 of depth 1 is applied several times. Each time the upper left block (low-level frequency coefficients) of a matrix of intermediate DAT-coefficients is transformed

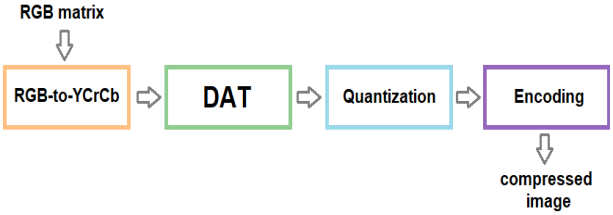


Fig. 8. Discrete atomic compression (DAC) of full-color 24-bit images

By design, the DAC is a lossy compression technique. This means that source and decompressed images are not identical. So, distortions are produced. However, there exists a quality loss control mechanism [26]. It is based on the properties of the atomic functions $\mathrm{up}_s(x)$ and provides an ability to get the desired level of distortion that is measured by the following classic metrics:

- Maximum absolute deviation (MAD):

$$\mathrm{MAD} = \max_i |X_i - Y_i|,$$

where $X = (X_1, \ldots, X_N)$ and $Y = (Y_1, \ldots, Y_N)$ are source and reconstructed data, respectively;

- Root mean square error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (X_i - Y_i)^2};$$

- Peak signal-to-noise ratio (PSNR):

$$\mathrm{PSNR} = 20\log_{10}(255/\mathrm{RMSE}).$$

In DAC, both DAT1 and DAT2 can be used. In [26], a comparison of their compression efficiency was conducted. It was shown that these schemes provided nearly the same compression ratio (CR) for any PSNR in the range from 35 dB to 48 dB. In this case, distortions are invisible to the human eye, despite the produced loss of quality.

The DAC algorithm can be modified to ensure lossless image compression. To achieve this feature, in [21], it is proposed to add encoded differences of source and decompressed images to a file with compressed data.

The convenience of DAC is ensured by the constructive properties of the atomic wavelet system (5). The locality of its functions provides quite small errors and, therefore, the reduction of distortions, as well as linear time complexity of the method. And, most importantly, the good approximation properties of (5) guarantee the data compression feature of DAC.

DAT1 and DAT2 are based on the same array transform. Nevertheless, these procedures have several major dissimilarities. The main difference is that DAT1 provides deeper decomposition of a source matrix. In general, it uses a higher number of functions from the system of atomic wavelets (5) and produces more blocks of DAT-coefficients than DAT2. By construction, the scheme DAT2 belongs to classic discrete wavelet transforms [27], which have numerous applications in image processing [20, 31].

In a sense, DAT1 provides “richer” image representation than DAT2 due to the higher number of feature levels. This property improves flexibility, as will be shown below, and might be useful in ML/CV.

Further, both DAT1 and DAT2 have asymptotically equivalent time complexity, which is of linear order with respect to the number of pixels of the processed image. However, our experience shows that DAT1 performs faster. This advantage is attributed to the organization of the memory hierarchy of computational architectures [32]. Indeed, the scheme DAT1 uses more localized matrix data that ensures better utilization of memory caches. In addition, DAT1 possesses better capabilities in terms of parallel computing [33].

Given the combination of advantages of DAT1, we focus on this scheme of DAT in the current research. Consider it in more detail.

As mentioned above, DAT1 transforms a given matrix M to $\Omega = (\Omega_{ik})$ that consists of $(m+1)(n+1)$ blocks. Each array DAT is based on the atomic function $\mathrm{up}_s(x)$, where s is a fixed positive integer. Therefore, DAT1 has three hyperparameters that define its settings.

When processing images with c channels, a matrix specifying each channel is transformed by DAT1 independently, and c matrices of DAT-coefficients are obtained. The computed matrices can be combined into a c-

channel matrix Ω. It represents atomic wavelet features of a source image. We call them **atomic embeddings**.

Atomic embeddings of images possess the following features:

– **Unsupervisedness**. DAT1 is a matrix function. This mapping has no unknown parameter that requires fitting. There is no need for training. From this point of view, DAT1 outperforms auto-encoders [34] that have recent applications to data compression [35], especially synthetic aperture radar [36] and satellite images [37];

– Low complexity. Time complexity of the scheme DAT1 is of the order O(N), where $N = hwd$ and h, w, and d are, respectively, height, width, and number of channels of a processed image. In terms of computational complexity, DAT1 is better than discrete cosine transform (DCT), which is a classic image processing tool [20, 27]. Indeed, the time complexity of DCT is of the order $O(N^2)$. However, in some partial cases, fast numerical schemes provide $O(N \cdot \log(N))$ [27]. Anyway, DAT has a lower time complexity than DCT;

– **Entire-image transformation capability**. DAT1 does not require splitting its input into blocks and can be applied to an entire image. This distinguishes DAT1 from DCT. Due to fundamental properties of trigonometric functions, in many applications, the block-splitting procedure precedes DCT, which is further applied to each obtained block separately. It is clear that the computed DCT-coefficients could be grouped with respect to their frequencies, providing an entire image representation. Meanwhile, this additional step increases the time of processing and might complicate software implementations;

– **Features grouping and shape-preserving**. By construction, each block Ω_{ik} of DAT1's output contains certain features of an entire input image. Ω_{00} has the smallest size. It contains an aggregation of source data. A downscaled copy of a processed image can be reconstructed using this block. For example, Figure 9 shows a sample satellite image provided by the European Space Agency (https://www.esa.int), and Figure 10 presents an image reconstructed from the block Ω_{00}. As it can be seen, Ω_{00} provides a source image preview. It preserves the shape of many objects and some of their features, especially colors. Nevertheless, better reconstruction of initial data requires other blocks Ω_{ik}.

Atomic embeddings could be useful in various image analysis tasks. They ensure compact representation of image features, and, therefore, their application could improve the performance of the existing ML and CV algorithms. It is clear that proving this hypothesis requires extensive research.

In the next Section, we consider the k-means clustering algorithm, which is a classic unsupervised learning method, and show that its performance can be increased using DAT1.

Fig. 9. Sample European Space Agency satellite image. Original: 3556 x 3486 pixels, 24-bit, 35.4 MB (raw)

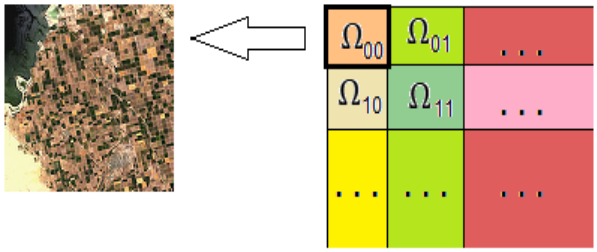


Fig. 10. Reconstruction of an image shown in Figure 9 using the blocks Ω_{00} (coefficients of the lowest frequency) of DAT-coefficients. A true-size image is displayed: 113 x 110 pixels, 24-bit, 36.5 KB (raw)

4. Results and Discussion

4.1. K-Means Clustering Using Atomic Embeddings

Pixel clustering splits an image into groups or clusters of pixels that have similar features, such as color, spatial location, and so on [20, 25]. It reduces the complexity of an input image and makes its further analysis more efficient. In some cases, it is used as a preprocessing step of higher-level techniques.

In this research, we consider the k-means clustering algorithm, which is a classic unsupervised learning method. It takes a set of vectors $I = \{p_1, p_2, \ldots, p_N\}$ and partitions it into k disjoint clusters $C = \{C_1, \ldots, C_k\}$ that minimize the following objective function:

$$J(C) = \sum_{j=1}^{k} \sum_{p \in C_j} |p - m_j|^2, \qquad (7)$$

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

where m_j is a mean vector of the set C_j and $|\cdot|$ denotes a norm. Vectors $\{m_j\}$ are called centroids. Note that k is a hyperparameter of the method.

Finding the minimum of the function (7) belongs to NP-hard problems. A wide range of heuristic methods providing an approximate solution have been recently developed [38]. The following approach is considered the "standard" k-means clustering algorithm:

- **Initialization**. Set an initial set of centroids $\{m_1,...,m_k\}$.
- **Assignment**. Assign each vector p_j to the cluster set whose centroid is the closest.
- **Update**. Update centroids: for each j=1,2, ..., k,

$$m_j = \frac{1}{n_j} \sum_{p \in C_j} p,$$

where n_j is the number of elements of the cluster C_j.

- **Test.** Compute the sum of norms of the differences between the corresponding current and previous centroids. If it is greater than the predefined threshold, then go back to step 2. Else, stop computations.

Time complexity of the iterative part can be presented as follows:

$$T = O(Ndki), \quad (8)$$

where N and d are, respectively, the number of clustered samples and their dimensionality, k is the number of clusters, and i is the number of iterations.

We note that the result significantly depends on the initial values chosen for the centroids $\{m_1, \dots, m_k\}$. In applications, the most common approach is k-means++ [39]. It selects centroids using sampling based on an empirical probability distribution of the vectors $I = \{p_j\}$. This initialization is slower than the use of random picking, but it ensures better clustering in combination with faster convergence.

In pixel clustering, vectors p_j represent points in a color space. Currently, we consider 24-bit full-color images. So, p_j is a vector of RGB-components (r, g, b), where each component is an integer from the range $(0,1,\dots,255)$. Note that normalization is usually applied before performing clustering.

Further, we propose two modifications of the k-means clustering algorithm that utilize atomic embeddings of images and explore the performance of the suggested approaches.

4.2. The K-Means Algorithm Modifications

Consider pixel clusterization of a given d-channel image with N pixels. Time complexity of the iterative part of the standard k-means algorithm is given by the asymptotic formula (8). It follows that if k and d, which specify the number of clusters and the dimensionality, respectively, are fixed, then a reduction of computations can be achieved by reducing the following items:

- **Iterations**. Indeed, a smaller value of i ensures fewer iterations. The proper initialization of the centroids $\{m_1, ..., m_k\}$ might accelerate the convergence and, therefore, reduce the required number of iterations.
- **Samples**. The number of analyzed pixels (N) is the major factor upon which the overall time complexity depends. However, to reduce it, major modifications of the clustering algorithm are needed.

We suggest the use of the k-means clustering method in combination with atomic embeddings of digital images. We propose to apply this algorithm to an image preview that is reconstructed using the block Ω_{00}, and, further, utilize the computed centroids for analysis of a source image. Figures 9 and 10 illustrate this idea. Indeed, the image preview shown in Figure 10 contains a significant amount of the color features of the source image given in Figure 9 and consists of fewer pixels. It is expected that preview analysis is faster, and its results are useful for analyzing a source image. It is clear that some modifications of the algorithm are required. We present them below.

Our first suggestion is to use image preview centroids as initializations for the algorithm that clusters the corresponding source image. The following steps provide the details:

- **Preview clustering**. Cluster a preview of a given image using k-means++ and find the centroids $\{\mu_1, \dots, \mu_k\}$ of the obtained clusters.
- **Source clustering.** Cluster a given source image using the following approach: set initial centroids to $\{\mu_1, \dots, \mu_k\}$, which are computed at the previous step, and perform the standard k-means method.

We call this algorithm k-means clustering with initialization by image preview centroids and denote it by **k-means (I)**.

Another proposition is to cluster a given source image directly by centroids of its preview:

- **Preview clustering**. Cluster a preview of a given image using k-means++ and find the centroids $\{\mu_1, \dots, \mu_k\}$ of the obtained clusters.
- **Source clustering.** Cluster a given source image by $\{\mu_1, \dots, \mu_k\}$. In other words, the cluster of each pixel p is equal to $\arg\min_{j=1,\dots,k} |p - \mu_j|$.

We call this technique k-means with cluster prediction by image preview centroids and denote it by **k-means (P)**. It is clear that this method is a partial case of the previous approach. Indeed, k-means (P) is the k-means (I) with a single iteration at the second step. Hence, it requires fewer computations and, thus, is faster.

Both modifications are expected to be less time-consuming than the classic k-means algorithm. In general, they produce different clustering results. So, their

comparison should be performed using proper numerical indicators. Further, we introduce them.

4.3. Metrics

To explore the efficiency of the proposed modifications of the k-means clustering method, we suggest several metrics that evaluate time performance and the difference between clusters provided by the considered methods.

Denote by $T_A(k)$ the time of clustering a given image using the algorithm A, where k is the number of clusters and A is one of the following algorithms: k-means++, k-means (I), or k-means (P). Consider the following acceleration ratios (AR):

$$AR_I(k) = T_{k-means++}(k) \,/\, T_{k-means\ (I)}(k), \qquad (9)$$

$$AR_P(k) = T_{k-means++}(k) \,/\, T_{k-means\ (P)}(k). \qquad (10)$$

The indicators (9) and (10) measure, respectively, how much faster the algorithms k-means (I) and k-means (P) run compared to k-means++, which is considered as a standard (reference).

Now, we introduce a criterion that measures clusters' similarity.

Let $I = \{p_1, p_2, \dots, p_N\}$ be a set of pixels of a given image. Applying the k-means++ algorithm to I computes centroids of clusters $M = \{m_1, \dots, m_k\}$ and the clusterization map $C = \{c_1, c_2, \dots, c_N\}$, where c_j specifies the cluster of the pixel p_j for any $j = 1,2, \dots, N$.

Further, let $\Lambda = \{\lambda_1, \dots, \lambda_k\}$ and $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_N\}$ be cluster centroids and the clusterization map provided by applying the k-means (I) to the same image I.

To evaluate the difference between C and Σ, the correspondence between clusters, which are produced by k-means++ and k-means (I), is required. We construct it as follows:

$$\rho(i) = \arg\min_{j=1,\dots,k} |\lambda_i - m_j|, \qquad (11)$$

where $i = 1, \dots, k$.

Next, let us apply the mapping (11) to each element of Σ. We obtain $\Psi = \{\psi_1, \psi_2, \dots, \psi_N\}$, where $\psi_j = \rho(\sigma_j)$, $j = 1,2, \dots, N$.

Now, when a proper correspondence between clusters is built, we introduce clusterization **similarity** that is defined by the formula:

$$Sim_{k-means\ (I)}(k) = \frac{T}{N} \cdot 100\%, \qquad (12)$$

where T is the number of elements of the set $\Pi = \{j \in \{1, \dots, N\}: \psi_j = c_j\}$.

The indicator (criterion) (12) measures the percentage of similarity between the clusterization maps provided by the k-means++ and k-means (I) algorithms.

Clusterization similarity $Sim_{k-means\ (P)}(k)$ between the clusterization maps produced by k-means++ and k-means (P) is defined in the same way.

Figure 11 shows the results of the application of k-means++, k-means (I), and k-means (P) techniques to the image given in Figure 9 with $k = 10$. Visually, the clusterization maps can be considered as identical (see Figures 11 (b), 11 (c), and 11 (e)), but there are dissimilarities marked with bright color (see data in Figures 11 (d) and (f)). Also, $Sim_{k-means\ (I)}(10) = 94\%$ and $Sim_{k-means\ (P)}(10) = 86\%$, which are equal to the percentage of "bright" pixels presented in Figure 11 (d) and (f), respectively.

The results shown in Figure 11 have been obtained using the Scikit-learn library [40]. Computations have been performed on AMD Ryzen 5 5600H 3.30 GHz CPU. The following values of ARs have been computed: $AR_I(10) = 2.47$ and $AR_P(10) = 24.24$.

This example shows that the proposed modifications of the k-means algorithm ensure nearly the same clustering while significantly reducing the computational time. A deeper exploration is given below.

4.4. Test Data Processing

Now, we compare the k-means++ method to k-means (I) and k-means (P) in terms of the proposed metrics. We conduct exploration using a set of different full-color 24-bit images. The following procedure is applied to each sample (test image):

- **Preview construction**. Compute atomic embeddings Ω_{00} for a given source image and reconstruct an image preview;
- **Clustering.** Perform k-means++ method, k-means (I) and k-means (P);
- **Evaluation**. Compute $Sim_{k-means\ (P)}(k)$, $Sim_{k-means\ (I)}(k)$, $AR_I(k)$ and $AR_P(k)$.

In this research, we use $k = 2,3, \dots, 20$. Atomic embeddings are provided by DAT1 based on the DAT of depth 5 and the function $up_{32}(x)$.

In addition to the proposed metrics, we also evaluate the silhouette coefficient (SC) [41]. This indicator provides numerical interpretation and validation of consistency within clusters of data. It varies in the range [-1,1]. The higher the value of SC, the better the clustering.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

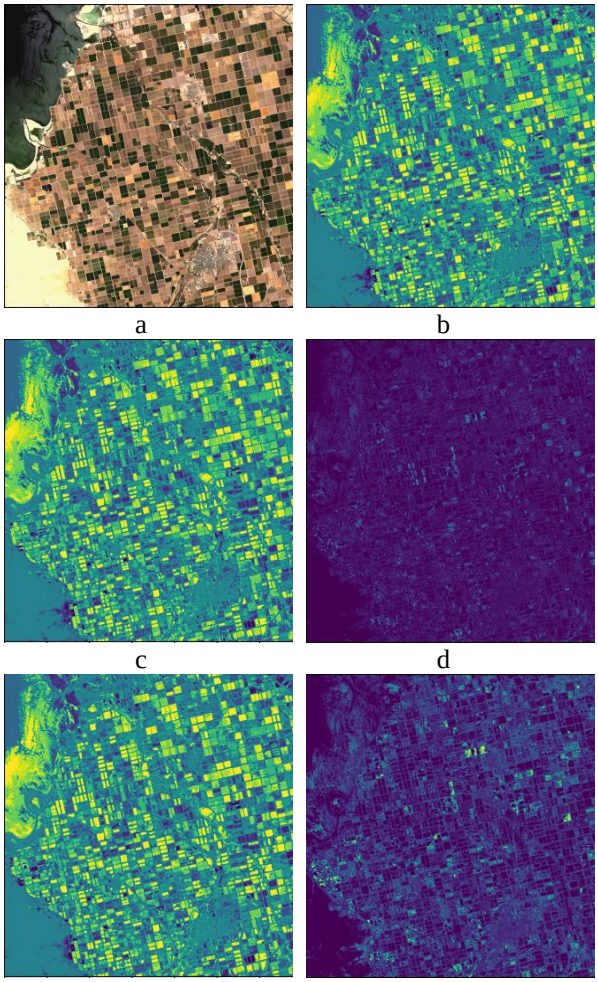


Fig. 11. Clustering the image shown in Figure 9 using the k-means++, k-means (I) and k-means (P) algorithms with k = 10: (**a**) Source image; (**b**) Clusterization map by k-means++; (**c**) Clusterization map by k-means (I); (**d**) The difference between clusterization maps produced by k-means++ and k-means (I) (points with "identical clusters" are dark); (**e**) Clusterization map k-means (P); (**f**) The difference between clusterization maps produced by k-means++ and k-means (P). In this case, $\mathrm{Sim}_{\mathrm{k-means\ (I)}}(10) = 94\%$ and $\mathrm{Sim}_{\mathrm{k-means\ (P)}}(10) = 86\%$

Moreover, we perform the same analysis of clustering using discrete wavelet transform (of depth 5) based on the Haar wavelet [27].

All computations are performed using the Scikit-learn library [40] on AMD Ryzen 5 5600H 3.30 GHz CPU.

We start with a sample image offered by the European Space Agency (ESA). Its scaled copy is given in Figure 9. The mage preview obtained using atomic embeddings is shown in Figure 10. The sizes of the source and the preview images are 3556 x 3486 and 113 x 110 pixels, respectively.

Table 1 presents the results of the clustering using the k-means (I) method. Also, it shows values of the indicator SC provided by the application of k-means++. Also, Figures 12, 13 and 14 compare the behavior of $\mathrm{Sim}_{\mathrm{k-means\ (I)}}$, $\mathrm{AR_I}$ and SC with respect to the number of clusters (k) growth.

Table 1

Clustering the image shown in Figure 9 using k-means (I)

k	Similarity, %		Acceleration rate		Silhouette coefficient		
	DAT	Haar	DAT	Haar	Kmeans ++	DAT	Haar
2	99	99	1.43	1.60	0.59	0.58	0.56
3	99	99	1.99	1.89	0.61	0.60	0.60
4	96	98	2.43	2.18	0.55	0.52	0.51
5	94	96	1.81	2.37	0.50	0.51	0.51
6	98	98	2.30	1.85	0.46	0.45	0.46
7	85	85	2.54	2.01	0.43	0.43	0.44
8	99	99	1.78	1.40	0.39	0.43	0.43
9	83	90	2.33	2.38	0.43	0.44	0.40
10	94	75	2.48	2.11	0.40	0.41	0.39
11	92	67	1.83	1.68	0.40	0.42	0.36
12	73	92	2.75	1.14	0.35	0.35	0.34
13	73	76	2.40	2.41	0.38	0.40	0.32
14	85	82	3.39	1.53	0.38	0.32	0.34
15	98	90	2.19	1.44	0.37	0.34	0.33
16	80	78	1.92	1.82	0.34	0.34	0.36
17	85	78	3.99	2.25	0.32	0.34	0.33
18	81	74	3.25	2.47	0.32	0.35	0.33
19	77	71	3.48	3.58	0.34	0.35	0.36
20	67	68	2.20	3.57	0.34	0.33	0.32

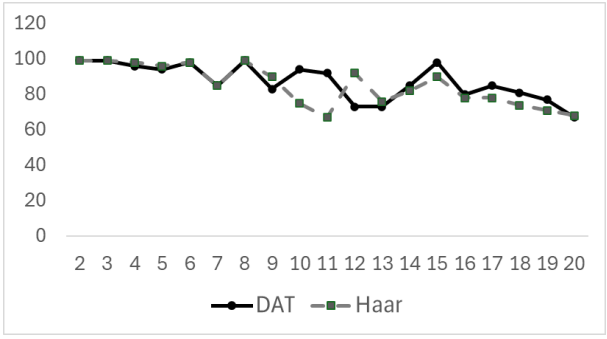


Fig. 12. Clustering the image shown in Figure 9 using k-means (I): the dependence of the similarity indicator $\mathrm{Sim}_{\mathrm{k-means\ (I)}}$ on the number of clusters k

Further, the results of the clustering using k-means (P) are given in Table 2. Figures 15-17 visualize them.

The given Tables are available at the following link to Google Drive folder: https://drive.google.com/drive/folders/115fGW8rcD9fJvNtAn7Trq_E0TdTx82Jf?usp=drive_link.

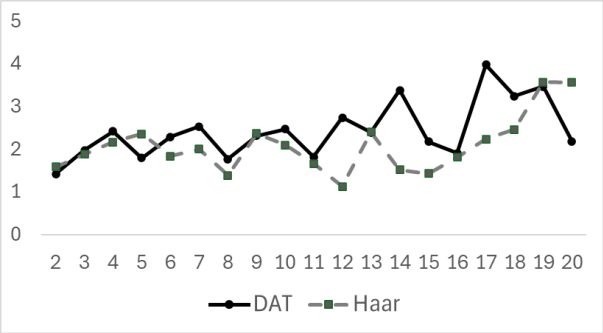


Fig. 13. Clustering the image shown in Figure 9 using k-means (I): the dependence of the acceleration rate AR_I on the number of clusters k

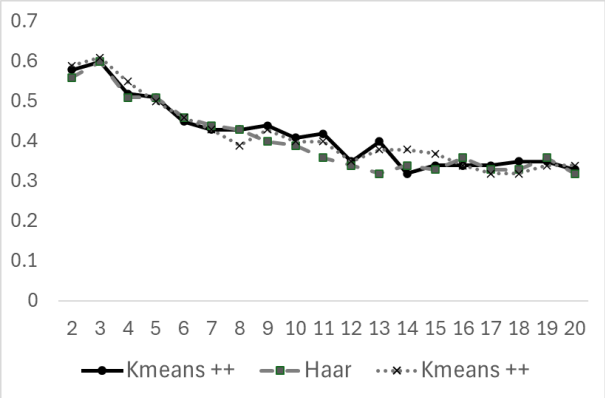


Fig. 14. Clustering the image shown in Figure 9 using k-means (I): the dependence of the silhouette coefficient SC on the number of clusters k

Table 2

Clustering the image shown in Figure 9 using k-means (P)

k	Similarity, %		Acceleration rate		Silhouette coefficient		
	DAT	Haar	DAT	Haar	Kmeans ++	DAT	Haar
2	97	97	9.76	10.84	0.59	0.57	0.57
3	98	96	11.27	10.14	0.61	0.59	0.53
4	95	94	17.19	16.86	0.55	0.51	0.50
5	87	90	20.18	18.21	0.50	0.50	0.50
6	96	86	16.20	15.01	0.46	0.49	0.43
7	81	76	23.33	22.51	0.43	0.48	0.46
8	84	75	31.45	29.23	0.39	0.42	0.42
9	77	79	27.37	25.38	0.43	0.44	0.42
10	86	75	24.24	22.59	0.40	0.38	0.39
11	82	80	23.72	24.63	0.40	0.39	0.37
12	75	76	33.06	32.56	0.35	0.36	0.37
13	78	69	30.02	29.29	0.38	0.38	0.33
14	83	75	30.65	30.48	0.38	0.35	0.34
15	83	76	35.75	33.55	0.37	0.34	0.31
16	81	76	41.08	38.94	0.34	0.32	0.34
17	82	67	40.58	39.45	0.32	0.35	0.32
18	81	68	45.17	42.36	0.32	0.35	0.33
19	75	67	46.52	45.29	0.34	0.33	0.29
20	73	63	47.23	45.84	0.34	0.30	0.31

Consider also a sample image from the Land-Cover.ai dataset [42]. Its scaled copy is shown in Figure 18. The original size is 8973 x 9429 pixels. It is an image of a very high resolution with a huge number of pixels. Figure 19 presents its preview obtained using atomic embeddings. One can see that the preview preserves many features that might be useful for pixel clustering the source image.

Tables 3 and 4 present the result of clusterization of the considered image by k-means++, k-means (I) and k-means (P). Figures 20-25 provide their visualization. This data are available at the following link to Google Drive folder: https://drive.google.com/drive/folders/1rP12eB1QTW8vgZM4iveyEor-Zobp_an59?usp=drive_link.

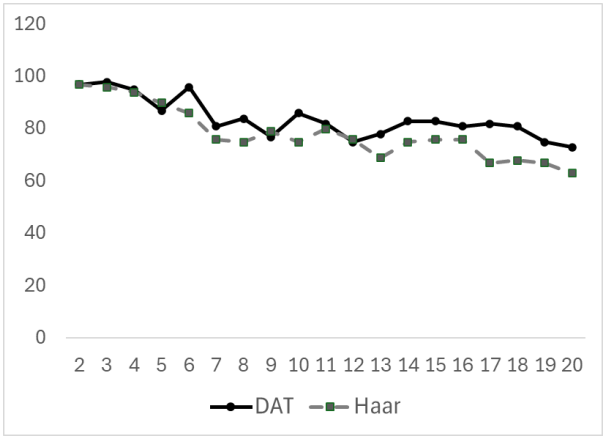


Fig. 15. Clustering the image shown in Figure 9 using k-means (P): the dependence of the similarity indicator $Sim_{k-means\ (P)}$ on the number of clusters k

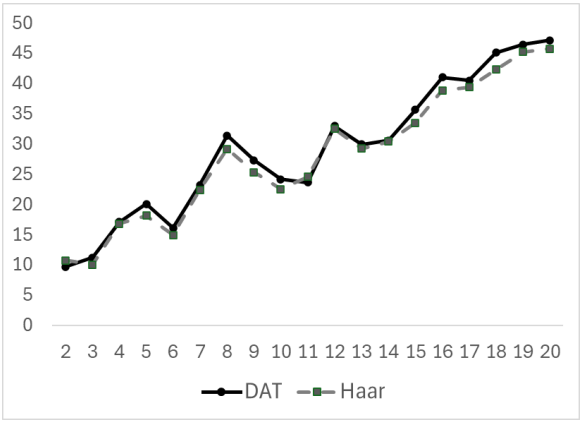


Fig. 16. Clustering the image shown in Figure 9 using k-means (P): the dependence of the acceleration rate AR_P on the number of clusters k

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

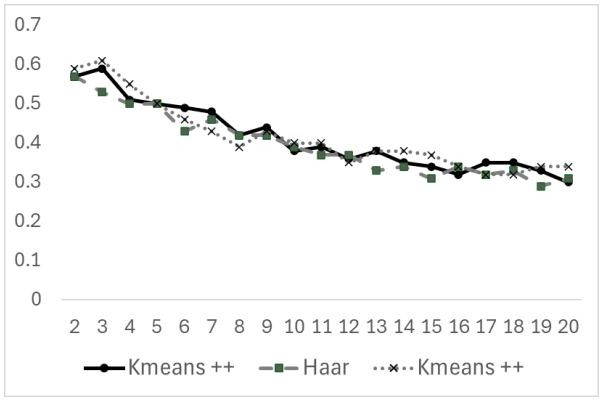


Fig. 17. Clustering the image shown in Figure 9 using k-means (P): the dependence of the silhouette coefficient SC on the number of clusters k

Fig. 18. Sample image from the LandCover.ai dataset (scaled copy is displayed). Original: 8973 x 9429 pixels, 24-bit, 242 MB (raw)

Fig. 19. Reconstruction of an image shown in Figure 18 using Ω_{00}. A true-size image is displayed: 282 x 296 pixels, 24-bit, 245 KB (raw)

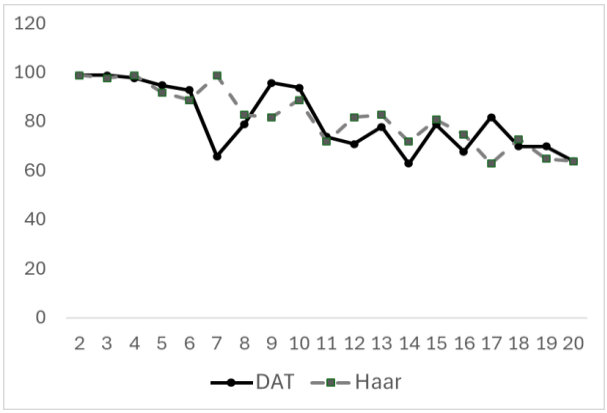


Fig. 20. Clustering the image shown in Figure 18 using k-means (I): the metric $\mathrm{Sim}_{\mathrm{k-means\ (I)}}$

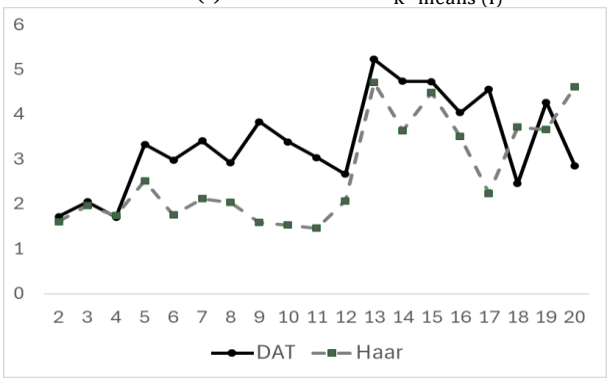


Fig. 21. Clustering the image shown in Figure 18 using k-means (I): the metric $\mathrm{AR_I}$

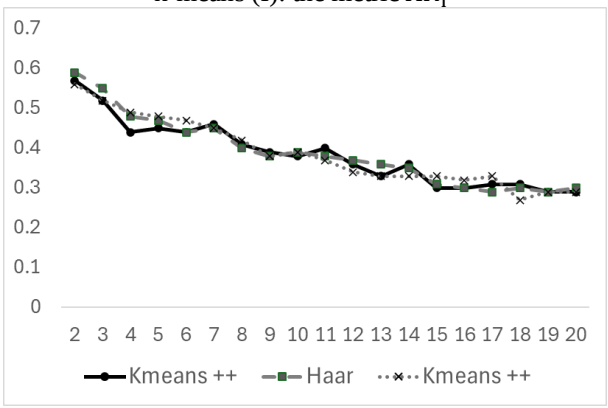


Fig. 22. Clustering the imageshown in Figure 18 using k-means (I): the indicator SC

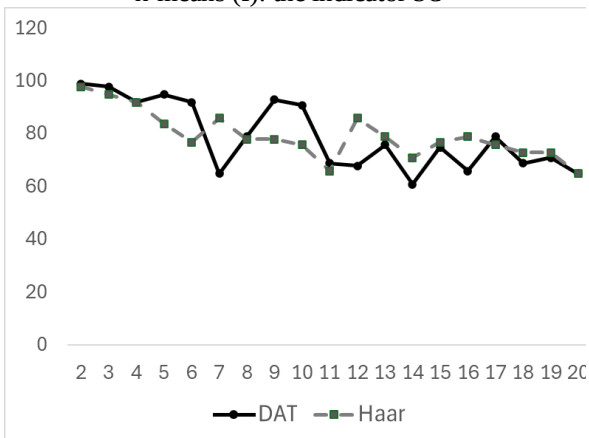


Fig. 23. Clustering the image shown in Figure 18 using k-means (P): the metric $\mathrm{Sim}_{\mathrm{k-means\ (P)}}$

Table 3

Clustering the image shown in Figure 18 by k-means (I)

k	Similarity, %		Acceleration rate		Silhouette coefficient		
	DAT	Haar	DAT	Haar	Kmeans ++	DAT	Haar
2	99	99	1.73	1.62	0.56	0.57	0.59
3	99	98	2.06	1.98	0.52	0.52	0.55
4	98	99	1.72	1.75	0.49	0.44	0.48
5	95	92	3.34	2.53	0.48	0.45	0.47
6	93	89	2.99	1.77	0.47	0.44	0.44
7	66	99	3.42	2.13	0.45	0.46	0.45
8	79	83	2.93	2.05	0.42	0.41	0.40
9	96	82	3.85	1.60	0.38	0.39	0.38
10	94	89	3.40	1.54	0.39	0.38	0.39
11	74	72	3.05	1.47	0.37	0.40	0.38
12	71	82	2.68	2.08	0.34	0.36	0.37
13	78	83	5.24	4.72	0.33	0.33	0.36
14	63	72	4.75	3.65	0.33	0.36	0.35
15	79	81	4.74	4.50	0.33	0.30	0.31
16	68	75	4.05	3.52	0.32	0.30	0.30
17	82	63	4.57	2.25	0.33	0.31	0.29
18	70	73	2.47	3.73	0.27	0.31	0.30
19	70	65	4.28	3.67	0.29	0.29	0.29
20	64	64	2.86	4.63	0.29	0.29	0.30

Table 4

Clustering the image shown in Figure 18 using k-means (P)

k	Similarity, %		Acceleration rate		Silhouette coefficient		
	DAT	Haar	DAT	Haar	Kmeans ++	DAT	Haar
2	99	98	14.23	15.10	0.56	0.58	0.61
3	98	95	18.55	15.36	0.52	0.52	0.51
4	92	92	17.50	15.20	0.49	0.50	0.50
5	95	84	27.35	22.71	0.48	0.47	0.45
6	92	77	21.54	19.28	0.47	0.42	0.44
7	65	86	26.94	23.00	0.45	0.43	0.40
8	79	78	37.88	24.71	0.42	0.35	0.41
9	93	78	38.43	26.68	0.38	0.40	0.37
10	91	76	42.36	28.11	0.39	0.40	0.34
11	69	66	33.05	27.48	0.37	0.36	0.33
12	68	86	44.51	30.74	0.34	0.35	0.32
13	76	79	49.91	37.61	0.33	0.37	0.31
14	61	71	51.44	38.29	0.33	0.36	0.31
15	75	77	50.83	37.84	0.33	0.31	0.32
16	66	79	47.61	37.6	0.32	0.32	0.30
17	79	76	42.46	35.10	0.33	0.31	0.28
18	69	73	49.85	40.02	0.27	0.30	0.29
19	71	73	49.59	37.30	0.29	0.31	0.29
20	65	65	60.55	46.98	0.29	0.26	0.29

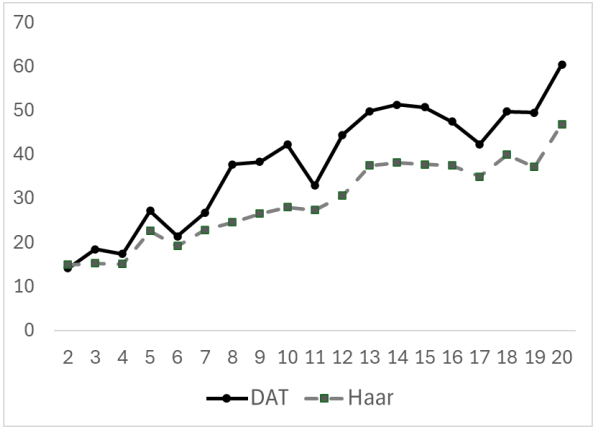


Fig. 24. Clustering the image shown in Figure 18 using k-means (P): the metric AR_P

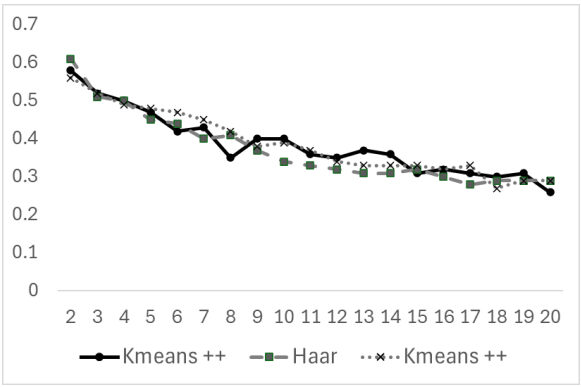


Fig. 25. Clustering the image shown in Figure 18 using k-means (P): the indicator SC

We also consider the USC-SIPI Aerials dataset available at the following site of the Signal and Image Processing Institute of the University of Southern California (USC-SIPI): https://sipi.usc.edu/database/ database.php?volume=aerials. Images from this dataset are smaller than the samples considered above. Figure 26 shows the largest of them, which is a 2250 x 2250 pixels image. Figure 27 presents its preview. It is clear that the majority of image features are lost. Hence, the results of the evaluation of the similarity indicators and silhouette coefficients are of particular interest. Below, we present their aggregated values. All other results can be found at the following link: https://drive.google.com/drive/ folders/1NyB7TWCjTT2K4s02CB1k7mAM-FqB76G7H?usp=sharing.

Tables 5 and 6 contain the aggregated results of evaluation of clustering images from the USC-SIPI Aerials dataset using k-means++, k-means (I) and k-means (P). Figures 28-31 provide their visualization.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

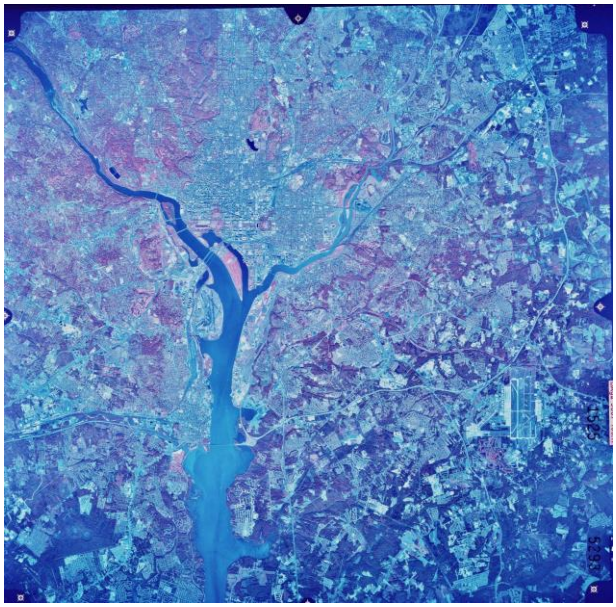


Fig. 26. Sample image from the Classic Aerials dataset (USC-SIPI Aerials). A scaled copy is displayed. Original image: 2250 x 2250 pixels, 24-bit, 14.48 MB (raw)

Fig. 27. Reconstruction of an image shown in Figure 26 using Ω_{00}. A true-size image is displayed: 72 x 72 pixels, 24-bit, 15.18 KB (raw)

4.5. Analysis

Analyzing the obtained results, we see the following. First, both k-means (I) and k-means (P) ensure high values of the evaluated similarity indicators $\mathrm{Sim}_{\mathrm{k-means\ (I)}}$ and $\mathrm{Sim}_{\mathrm{k-means\ (P)}}$, respectively, for any $k = 2,3,\ldots,20$ and each applied discrete transform. So, these modifications of the k-means algorithm provide clusters that are close to clusters produced by k-means++. Furthermore, there is a minor deviation between values of silhouette coefficients (see Figures 14, 17, 22, 25, 29, and 31), which means that the produced clusters have nearly the same quality measured by the indicator SC.

Besides, the proposed methods perform faster than k-means++, which is shown in Figures 13, 16, 21, and 24. Specifically, k-means (P) guarantees a significantly higher acceleration compared to k-means (I).

Second, an analysis of the behavior of the similarity metrics $\mathrm{Sim}_{\mathrm{k-means\ (I)}}$ and $\mathrm{Sim}_{\mathrm{k-means\ (P)}}$ shows that they cannot be considered homogeneous with respect to k. There is a significant dependence of these indicators on the content of the clustered image. However, a decreasing trend is obvious with an increase in the number of clusters, which is illustrated by Figures 28 and 30 that visualize aggregated data. At the same time, the silhouette coefficient demonstrates a more robust behavior (less decrease).

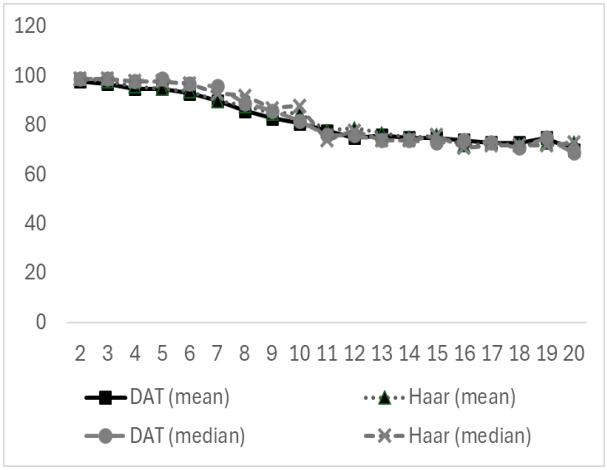


Fig. 28. Clustering images from the USC-SIPI Aerials dataset using k-means (I): the metric $\mathrm{Sim}_{\mathrm{k-means\ (I)}}$

Table 5

Clustering images from the USC-SIPI Aerials dataset using k-means (I)

k	Mean similarity, %		Median similarity, %		Mean Silhouette coefficient		
	DAT	Haar	DAT	Haar	DAT	Haar	Kmeans ++
2	98	99	99	99	0.57	0.57	0.57
3	97	98	99	99	0.51	0.49	0.50
4	95	96	98	98	0.45	0.44	0.45
5	95	95	99	98	0.42	0.43	0.42
6	93	94	97	97	0.40	0.40	0.39
7	90	90	96	93	0.38	0.38	0.37
8	86	88	89	92	0.36	0.36	0.36
9	83	85	86	87	0.35	0.34	0.34
10	81	85	82	88	0.33	0.33	0.34
11	78	78	76	74	0.32	0.32	0.32
12	75	79	76	78	0.31	0.31	0.32
13	76	77	74	74	0.30	0.30	0.31
14	75	75	74	74	0.30	0.30	0.30
15	75	76	73	76	0.29	0.30	0.30
16	74	72	74	71	0.29	0.28	0.29
17	73	73	73	72	0.28	0.28	0.29
18	73	72	71	72	0.28	0.28	0.29
19	75	73	75	72	0.28	0.27	0.28
20	70	73	69	73	0.27	0.27	0.28

Table 6

Clustering images from the USC-SIPI Aerials dataset using k-means (P)

k	Mean similarity, %		Median similarity, %		Mean Silhouette coefficient		
	DAT	Haar	DAT	Haar	DAT	Haar	Kmeans ++
2	93	93	95	95	0.56	0.56	0.57
3	88	82	93	88	0.50	0.48	0.50
4	82	76	86	77	0.46	0.42	0.45
5	79	74	80	75	0.41	0.41	0.42
6	79	71	82	72	0.39	0.37	0.39
7	78	73	80	74	0.37	0.34	0.37
8	76	70	78	70	0.35	0.33	0.36
9	72	70	73	72	0.33	0.32	0.34
10	71	69	71	70	0.31	0.30	0.34
11	70	67	70	70	0.31	0.29	0.32
12	67	65	67	65	0.30	0.27	0.32
13	67	65	68	66	0.29	0.27	0.31
14	66	64	66	65	0.28	0.26	0.30
15	66	62	65	64	0.27	0.25	0.30
16	66	62	66	64	0.27	0.25	0.29
17	65	61	66	63	0.27	0.24	0.29
18	63	61	64	63	0.25	0.23	0.29
19	63	61	65	61	0.26	0.22	0.28
20	62	61	65	62	0.26	0.23	0.28

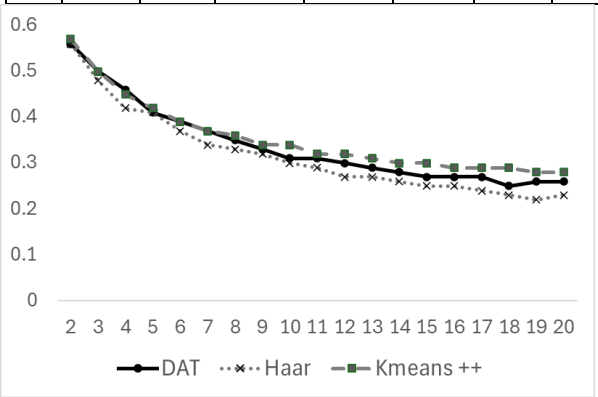


Fig. 29. Clustering images from the USC-SIPI Aerials dataset using k-means (I): the silhouette coefficient

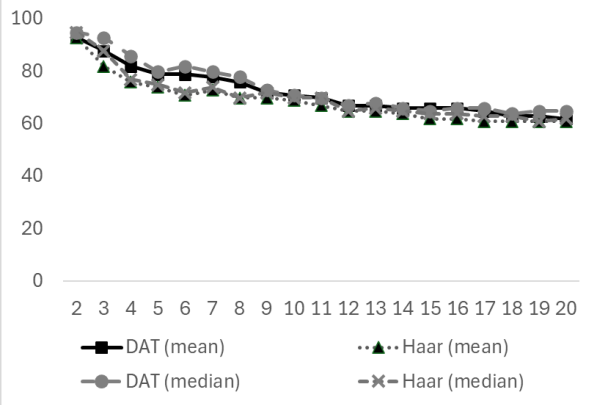


Fig. 30. Clustering images from the USC-SIPI Aerials dataset using k-means (P): the metric $\mathrm{Sim}_{\mathrm{k-means\ (P)}}$

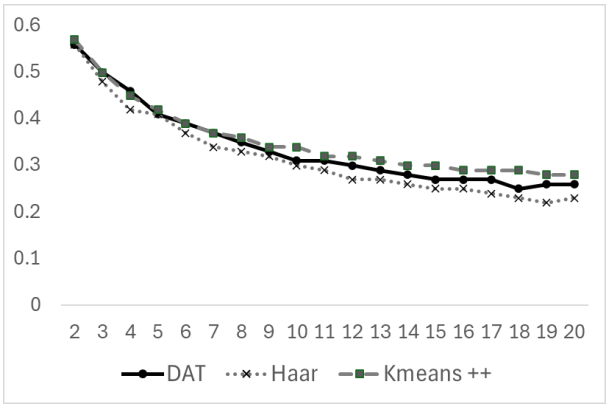


Fig. 31. Clustering images from the USC-SIPI Aerials dataset using k-means (P): the silhouette coefficient

Next, comparing the use of DAT- and Haar-based transforms, the following conclusions can be drawn:

– On average, DAT provides better results than discrete Haar wavelet transform with respect to the similarity metrics (see Figures 28, 30). However, the difference is not significant. Moreover, in some particular cases, Haar-based representation is more efficient;

– In most cases, DAT guarantees a higher value of the silhouette coefficient, but the difference is insignificant;

– Generally, DAT ensures better time performance, especially when it is applied in the scope of k-means (P) technique.

Despite the minor differences between the DAT- and Haar-based transforms in terms of the similarity measure, we have noticed that they produce non-coinciding clusters. If we compare them to clusters provided by k-means++, we notice the following: the Haar-based clustering produces greater distinctions at object boundaries, while distinctions produced by the DAT-based technique are more concentrated within objects. Figures 32 and 33 illustrate this phenomenon.

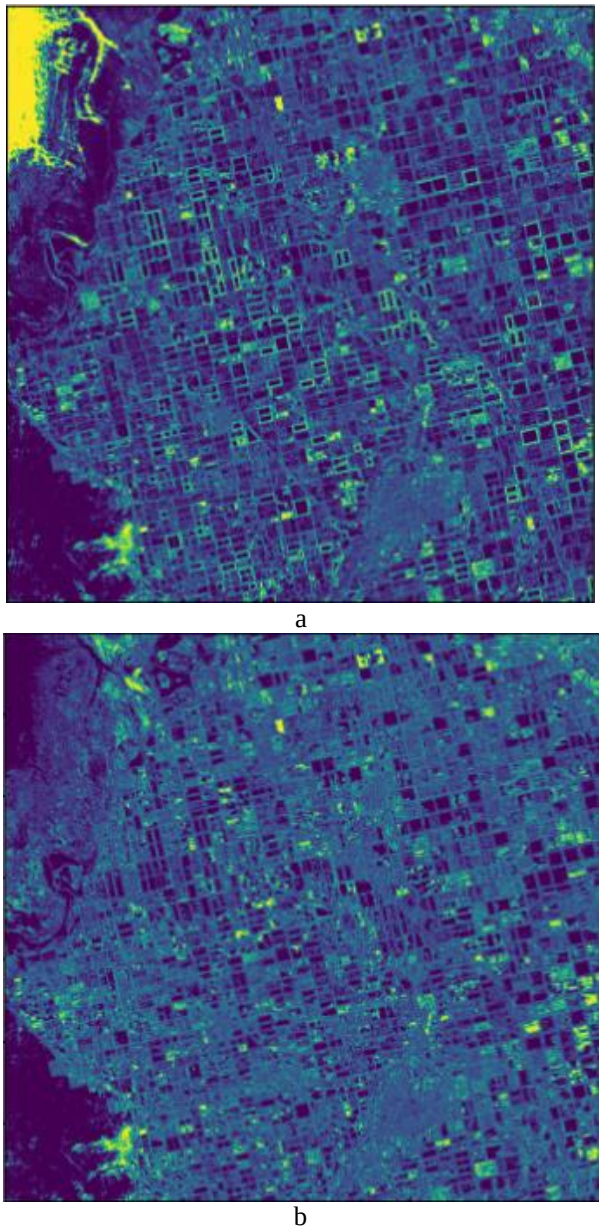
In the next Section, we further discuss the obtained results from different viewpoints. Also, we consider possible applications of the introduced methods and atomic embeddings of digital images.

4.6. Discussion

The proposed modifications of the k-means algorithm can be considered a special heuristic for the initialization of clusters' centroids. By design, the k-means (P) technique is the restricted version of k-means (P). Despite the simplicity of k-means (P), the results of the test data analysis show that it provides an acceptable approximation of clusters produced by the k-means++ method. This feature is due to good approximation properties of atomic functions and confirms the hypothesis that the image representation by atomic embeddings is ML/CV-oriented with respect to the k-means clustering.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

The highest difference between clusters, produced by the basic method and the suggested modifications, appears at object boundaries. This is a consequence of the aggregating features of the applied discrete transforms.

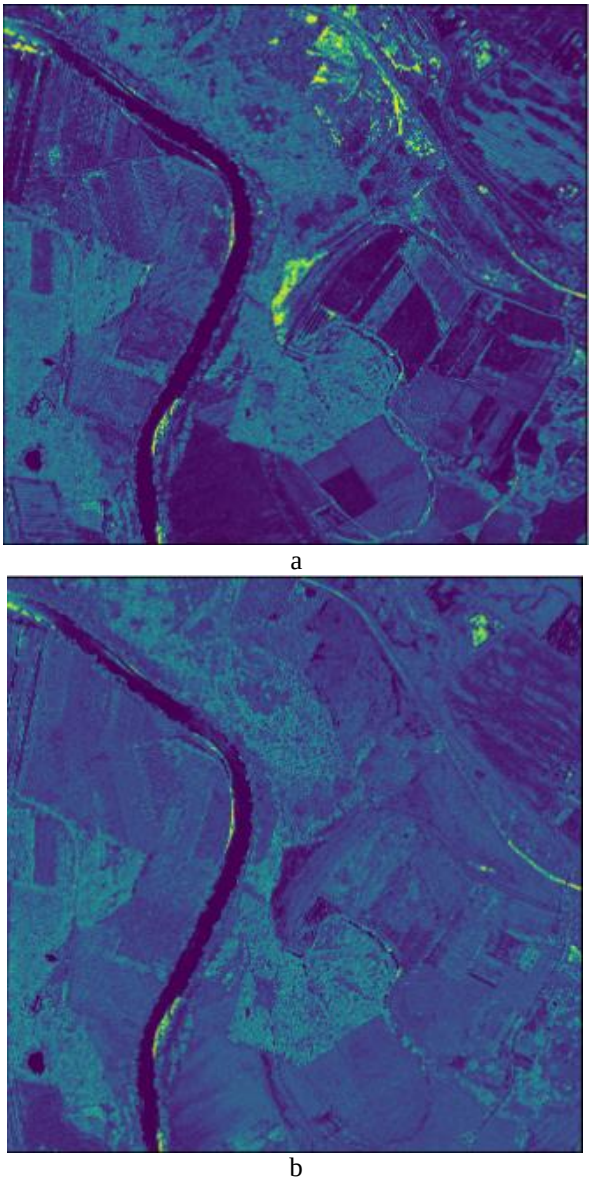


a

b

Fig. 32. The difference between clusters provided by k-means++ and k-means (P) for the image shown in Figure 9 for k = 12: (**a**) Haar-based clustering, $\mathrm{Sim}_{\mathrm{k-means\ (P)}}(12) = 76\%$; (**b**) DAT-based clustering, $\mathrm{Sim}_{\mathrm{k-means\ (P)}}(12) = 75\%$. The Haar-based clustering produces greater distinctions at object boundaries, while distinctions produced by the DAT-based technique are more concentrated within objects. Scaled images are given. Original size: 3556 x 3486 pixels

Indeed, in both k-means (I) and k-means (P), the wavelet coefficients corresponding to the lowest-level frequency band are applied. However, they contain poor or no information on object boundaries, especially if the depth of the wavelet transform is large. The use of a higher number of blocks of wavelet coefficients could reduce the deviation, but it would increase the size of the processed image preview and, hence, computations.



a

b

Fig. 33. The difference between clusters provided by the k-means++ and k-means (P) for the image shown in Figure 18 for k = 13: (**a**) Haar-based clustering, $\mathrm{Sim}_{\mathrm{k-means\ (P)}}(13) = 79\%$; (**b**) DAT-based clustering, $\mathrm{Sim}_{\mathrm{k-means\ (P)}}(13) = 76\%$. The Haar-based clustering produces greater distinctions at object boundaries, while distinctions produced by DAT-based technique are more concentrated within objects. Scaled images are given. Original size: 8973 x 9429 pixels

In this research, we use the proposed similarity measure and silhouette coefficient. Their main advantage is that these indicators do not require labeled data and can be evaluated in an unsupervised manner. Meanwhile, they do not assess the relationship between particular clusters. The intersection over union (IoU) metric might

be more suitable for this purpose [25]. This classic indicator is widely used in image segmentation analysis and provides the comparison of two clusters of pixels. So, IoU is more local than $\mathrm{Sim}_{\mathrm{k-means\ (I)}}$ and $\mathrm{Sim}_{\mathrm{k-means\ (P)}}$ that analyze the whole cluster maps. We plan to use it in the future.

Next, this paper considers unsupervised methods for image analysis and processing. In particular, the DAT transform produces image embeddings, eliminating the need for any model training. We apply them in combination with only one of the wide range of clustering algorithms. We expect that the use of atomic embeddings in other methods might improve their efficiency.

Further, the matrix transform DAT1 has flexible settings. By varying its hyperparameters, including the depth of the array DAT, one may get the desired customization. Moreover, various mixtures of the atomic functions $\mathrm{up}_s(x)$ can be used in this transform. Besides, other function systems, for instance, generalized atomic wavelets, can be applied for the construction of image representation that is similar to the one considered in this paper [43].

Finally, the obtained results show a slight advantage of atomic wavelets over the Haar wavelet. The difference is insignificant since a tiny subset of wavelet coefficients is applied (currently, the ratio is about 1 to 1024). It is expected that using more coefficients will make this advantage more noticeable due to approximation properties of spaces of atomic functions [23].

5. Conclusions

In this research, we have considered image representation, which is built using the atomic functions $\mathrm{up}_s(x)$ and called atomic embeddings, from the image analysis perspective. Discrete atomic transform was earlier developed mainly for solving image compression tasks [26, 29]. It is the core of the DAC algorithm, which provides both compression and encryption features [21]. In this paper, we have proposed applying atomic embeddings to solve machine learning and computer vision tasks.

The potential of the proposed approach has been demonstrated with respect to the k-means clustering algorithm. This classic unsupervised learning technique is widely applied as a part of complex image analysis pipelines [44] aimed to solve applied problems of innovate agriculture [45, 46] and medicine [47, 48].

We have introduced two modifications of the k-means algorithm. Both proposed approaches leverage a minor part of atomic embeddings. Their efficiency has been evaluated using acceleration ratio and clusterization map similarity metrics. The analysis of test remote sensing images has demonstrated a significantly faster approach to achieving nearly the same clustering results. Hence, image representation by atomic functions $\mathrm{up}_s(x)$ is ML/CV-oriented in terms of k-means clustering. Meanwhile, we expect that image representation by atomic functions $\mathrm{up}_s(x)$ might be useful in a considerably wider range of ML/CV methods. In particular, it could be useful in semantic and instance segmentation [25, 44], where the proposed modifications of k-means could be performed as a preprocessing step. This can be one of the tasks to be solved in the future.

We have also compared the atomic and Haar wavelets. It follows that, in general, atomic wavelets perform better. However, the obtained results do not provide a comprehensive comparison in terms of their usability in image analysis. Moreover, other wavelet systems have not been considered. This aspect will be addressed in future research.

Contributions of authors: conception – **Vladimir Lukin**, **Viktor Makarichev**; methodology – **Vladimir Lukin**; problem formulation – **Vladimir Lukin, Viktor Makarichev**; analysis – **Viktor Makarichev, Vladimir Lukin**, **Sergii Kryvenko, Iryna Brysina**; model development – **Viktor Makarichev**; software – **Viktor Makarichev**; validation – **Vladimir Lukin**, **Sergii Kryvenko**; analysis of results – **Viktor Makarichev, Vladimir Lukin**, **Sergii Kryvenko, Iryna Brysina**; visualization – **Iryna Brysina**; writing –**Viktor Makarichev**; revision and editing – **Vladimir Lukin**, **Sergii Kryvenko, Iryna Brysina**.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship, or otherwise, that could affect the research and its results presented in this paper.

Acknowledgments

The authors are personally grateful to V.A. Rvachev for his attention and support of this research. The authors sincerely appreciate J. Koloda for inspiring the initiation of this research.

Financing

The study was conducted without financial support.

Data availability

The results of the analysis of the test data presented in the paper are given in the set of CSV files available at the following link to Google Drive folder: https://drive.google.com/drive/folders/1WA8YCiK-

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Xwq33Kxe5NtnWcWb9kIv5JDR?usp=sharing (accessed on 12 May 2025). The original data presented in the study are openly available in the following resources: https://sipi.usc.edu/database/database.php?volume=aerials (USC-SIPI Aerials, accessed on 12 May 2025), https://www.kaggle.com/datasets/adrianboguszewski/landcoverai (the LandCover.ai dataset, accessed on 12 May 2025), and https://www.esa.int/ESA_Multimedia (the European Space Agency, accessed on 12 May 2025). The results of processing and analyzing the USC-SIPI Aerials dataset are available at the following link to the Google Drive folder: https://drive.google.com/drive/folders/1NyB7TWCjTT2K4s02CB1k7mAM-FqB76G7H?usp=sharing (accessed on 12 May 2025). The results of the analysis of the image from ESA can be found at: https://drive.google.com/drive/folders/115fGW8rcD9fJvNtAn7Trq_E0TdTx82Jf?usp=drive_link (accessed on 12 May 2025). The results of the LandCover.ai image is available at https://drive.google.com/drive/folders/1rP12eB1QTW8vgZM4iveyEor-Zobp_an59?usp=sharing (accessed on 12 May 2025).

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

All authors have read and agreed to the published version of this manuscript.

References

1. Wang, R., Sun, Y., Zong, J., Wang, Y., Cao, X., Wang, Y., Cheng, X., & Zhang, W. Remote Sensing Application in Ecological Restoration Monitoring: A Systematic Review. *Remote Sensing*, 2024, vol. 16, article no. 2204. DOI: 10.3390/rs16122204.

2. Wasehun, E. T., Hashemi Beni, L., & Di Vittorio, C. A. UAV and satellite remote sensing for inland water quality assessments: a literature review. *Environmental Monitoring and Assessment*, 2024, vol. 196, article no. 277. DOI: 10.1007/s10661-024-12342-6.

3. Wang, J., Wang, Y., Li, G., & Qi, Z. Integration of Remote Sensing and Machine Learning for Precision Agriculture: A Comprehensive Perspective on Applications. *Agronomy*, 2024, vol. 14, iss. 9, article no. 1975. DOI: 10.3390/agronomy14091975.

4. Mehedi, I. M., Hanif, M. S., Bilal, M., Vellingiri, M. T., & Palaniswamy, T. Remote Sensing and Decision Support System Applications in Precision Agriculture: Challenges and Possibilities. *IEEE Access*, 2024, vol. 12, pp. 44786-44798. DOI: 10.1109/ACCESS.2024.3380830.

5. Koukiou, G. SAR Features and Techniques for Urban Planning—A Review. *Remote Sensing*, 2024, vol. 16, iss. 11, article no. 1923. DOI: 10.3390/rs16111923.

6. Al Shafian, S., & Hu, D. Integrating Machine Learning and Remote Sensing in Disaster Management: A Decadal Review of Post-Disaster Building Damage Assessment. *Buildings*, 2024, vol. 14, iss. 8, article no. 2344. DOI: 10.3390/buildings14082344.

7. Yang, Y., Ju, Y., Gao, Y., Zhang, C., & Lam, K.-M. Remote sensing insights into ocean fronts: a literature review. *Intelligent Marine Technology and Systems*, 2024, vol. 2, article no. 10. DOI: 10.1007/s44295-024-00024-5.

8. Ye, Q., Wang, Y., Liu, L., Guo, L., Zhang, X., Dai, L., Zhai, L., Hu, Y., Ali, N., Ji, X., et al. Remote Sensing and Modeling of the Cryosphere in High Mountain Asia: A Multidisciplinary Review. *Remote Sensing*, 2024, vol. 16, iss. 10, article no. 1709. DOI: 10.3390/rs16101709.

9. Kadhim, I., & Abed, F. M. A Critical Review of Remote Sensing Approaches and Deep Learning Techniques in Archaeology. *Sensors*, 2023, vol. 23, iss. 6, article no. 2918. DOI: 10.3390/s23062918.

10. Avtar, R., Kouser, A., Kumar, A., Singh, D., Misra, P., Gupta, A., Yunus, A. P., Kumar, P., Johnson, B. A., Dasgupta, R., et al. Remote Sensing for International Peace and Security: Its Role and Implications. *Remote Sensing*, 2021, vol. 13, iss. 3, article no. 439. DOI: 10.3390/rs13030439.

11. Sayood, K. *Introduction to Data Compression*, 5th ed. Morgan Kaufman: Cambridge, MA, USA, 2017.

12. Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 2017, vol. 14, no. 5, pp. 778-782. DOI: 10.1109/LGRS.2017.2681128.

13. Wang, L., Zhang, M., Gao, X., & Shi, W. Advances and Challenges in Deep Learning-Based Change Detection for Remote Sensing Images: A Review through Various Learning Paradigms. *Remote Sensing*, 2024, vol. 16, iss. 5, article no. 804. DOI: 10.3390/rs16050804.

14. Vasile, C.-E., Ulmămei, A.-A., & Bîră, C. Image Processing Hardware Acceleration – A Review of Operations Involved and Current Hardware Approaches. *Journal of Imaging*, 2024, vol. 10, iss. 12, article no. 298. DOI: 10.3390/jimaging10120298.

15. Iqbal, U., Davies, T., & Perez, P. A Review of Recent Hardware and Software Advances in GPU-Accelerated Edge-Computing Single-Board Computers (SBCs) for Computer Vision. *Sensors*, 2024, vol. 24, iss. 15, article no. 4830. DOI: 10.3390/s24154830.

16. Hua, H., Li, Y., Wang, T., Dong, N., Li, W., & Cao, J. Edge Computing with Artificial Intelligence: A

Machine Learning Perspective. *ACM Computing Surveys*, 2023, vol. 55, no. 9, article no. 184. DOI: 10.1145/3555802.

17. Jouini, O., Sethom, K., Namoun, A., Aljohani, N., Alanazi, M. H., & Alanazi, M. N. A Survey of Machine Learning in Edge Computing: Techniques, Frameworks, Applications, Issues, and Research Directions. *Technologies*, 2024, vol. 12, iss. 6, article no. 81. DOI: 10.3390/technologies12060081.

18. Shi, Y.-Q., & Sun, H. *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards*, 3rd ed. CRC Press, 2021.

19. Bull, D., & Zhang, F. *Intelligent Image and Video Compression: Communicating Pictures*, 2nd ed. Academic Press, 2021.

20. Gonzalez, R. C., & Woods, R. E. *Digital Image Processing*, 2nd ed. Prentice Hall: Upper Saddle River, NJ, USA, 2002.

21. Makarichev, V., Lukin, V., Illiashenko, O., & Kharchenko, V. Digital Image Representation by Atomic Functions: The Compression and Protection of Data for Edge Computing in IoT Systems. *Sensors*, 2022, vol. 22, iss. 10, article no. 3751. DOI: 10.3390/s22103751.

22. Rvachev, V. A. Compactly supported solutions of functional-differential equations and their applications. *Russian Mathematical Surveys*, 1990, vol. 45, no. 1, pp. 87-120. DOI: 10.1070/RM1990v045n01ABEH002324.

23. Makarichev, V. A. Approximation of periodic functions using mup_s (x). *Mathematical Notes*, 2013, vol. 93, pp. 858-880. DOI: 10.1134/S0001434613050258.

24. Makarichev, V., Lukin, V., & Brysina, I. On the Impact of Discrete Atomic Compression on Image Classification by Convolutional Neural Networks. *Computation*, 2024, vol. 12, iss. 9, article no. 176. DOI: 10.3390/computation12090176.

25. Szeliski, R. *Computer Vision: Algorithms and Applications*, 2nd ed. Springer: Cham, Switzerland, 2022.

26. Makarichev, V., Vasilyeva, I., Lukin, V., Vozel, B., Shelestov, A., & Kussul, N. Discrete Atomic Transform-Based Lossy Compression of Three-Channel Remote Sensing Images with Quality Control. *Remote Sensing*, 2022, vol. 14, iss. 1, article no. 125. DOI: 10.3390/rs14010125.

27. Chui, C. K., & Jiang, Q. *Applied Mathematics: Data Compression, Spectral Methods, Fourier Analysis, Wavelets, and Applications*. Atlantis Press: Paris, France, 2013.

28. Makarichev, V., & Kharchenko, V. Application of dynamic programming approach to computation of atomic functions. *Radioelectronic and Computer Systems*, 2021, no. 4, pp. 36-45. DOI: 10.32620/reks.2021.4.03.

29. Makarichev, V., Lukin, V., & Brysina, I. On the Applications of the Special Class of Atomic Functions: Practical Aspects and Perspectives. In *Integrated Computer Technologies in Mechanical Engineering*; Nechyporuk, M., Pavlikov, V., Kritskiy, D., Eds.; Springer: Cham, Switzerland, 2021; vol. 188, pp. 42-54. DOI: 10.1007/978-3-030-66717-7_4.

30. Berkolaiko, M., & Novikov, I. On infinitely smooth compactly supported almost-wavelets. *Mathematical Notes*, 1994, vol. 56, pp. 877-883. DOI: 10.1007/BF02362405.

31. Welstead, S. *Fractal and Wavelet Image Compression Techniques*. SPIE Publications: Bellingham, WA, USA, 1999.

32. Bryant, R. E., & O'Hallaron, D. R. *Computer Systems: A Programmer's Perspective*, 3rd ed. Pearson: London, UK, 2015.

33. Robey, R., & Zamora, Y. *Parallel and High Performance Computing*. Manning: NY, USA, 2021.

34. Chen, S., & Guo, W. Auto-Encoders in Deep Learning—A Review with New Perspectives. *Mathematics*, 2023, vol. 11, iss. 8, article no. 1777. DOI: 10.3390/math11081777.

35. Huang, C.-H., & Wu, J.-L. Unveiling the Future of Human and Machine Coding: A Survey of End-to-End Learned Image Compression. *Entropy*, 2024, vol. 26, iss. 5, article no. 357. DOI: 10.3390/e26050357.

36. Xu, Q., Xiang, Y., Di, Z., Fan, Y., Feng, Q., Wu, Q., & Shi, J. Synthetic Aperture Radar Image Compression Based on a Variational Autoencoder. *IEEE Geoscience and Remote Sensing Letters*, 2022, vol. 19, article no. 4015905. DOI: 10.1109/LGRS.2021.3097154.

37. Alves de Oliveira, V., Chabert, M., Oberlin, T., Poulliat, C., Bruno, M., Latry, C., Carlavan, M., Henrot, S., Falzon, F., & Camarero, R. Reduced-Complexity End-to-End Variational Autoencoder for on Board Satellite Image Compression. *Remote Sensing*, 2021, vol. 13, iss. 3, article no. 447. DOI: 10.3390/rs13030447.

38. Ahmed, M., Seraj, R., & Islam, S. M. S. The k-means Algorithm: A Comprehensive Survey and Performance Evaluation. *Electronics*, 2020, vol. 9, iss. 8, article no. 1295. DOI: 10.3390/electronics9081295.

39. Arthur, D., & Vassilvitskii, S. K-means++: the advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 7-9 January 2007, pp. 1027-1035.

40. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2011, vol. 12, pp. 2825-2830.

41. Kaufman, L., & Rousseeuw, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons: Hoboken, NJ, USA, 1990.

42. Boguszewski, A., Batorski, D., Ziemba-Jankowska, N., Dziedzic, T., & Zambrzycka, A. LandCover.ai: Dataset for Automatic Mapping of Buildings, Woodlands, Water and Roads from Aerial Imagery. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 20-25 June 2021, Nashville, TN, USA, pp. 1102-1110.

43. Brysina, I., & Makarichev, V. Generalized Atomic Wavelets. *Radioelectronic and Computer Systems*, 2018, no. 1, pp. 23-31. DOI: 10.32620/reks.2018.1.03.

44. Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., Yang, T., & Gao, M. Techniques and Challenges of Image Segmentation: A Review. *Electronics*, 2023, vol. 12, iss. 5, article no. 1199. DOI: 10.3390/electronics12051199.

45. Omia, E., Bae, H., Park, E., Kim, M. S., Baek, I., Kabenge, I., & Cho, B.-K. Remote Sensing in Field Crop Monitoring: A Comprehensive Review of Sensor Systems, Data Analyses and Recent Advances. *Remote Sensing*, 2023, vol. 15, iss. 2, article no. 354. DOI: 10.3390/rs15020354.

46. Orchi, H., Sadik, M., & Khaldoun, M. (2022). On Using Artificial Intelligence and the Internet of Things for Crop Disease Detection: A Contemporary Survey. *Agriculture*, 2022, vol. 12, iss. 1, article no. 9. DOI: 10.3390/agriculture12010009.

47. Arabahmadi, M., Farahbakhsh, R., & Rezazadeh, J. Deep Learning for Smart Healthcare – A Survey on Brain Tumor Detection from Medical Imaging. *Sensors*, 2022, vol. 22, iss. 5, article no. 1960. DOI: 10.3390/s22051960.

48. Xu, Y., Quan, R., Xu, W., Huang, Y., Chen, X., & Liu, F. Advances in Medical Image Segmentation: A Comprehensive Review of Traditional, Deep Learning and Hybrid Approaches. Bioengineering, 2024, vol. 11, iss. 10, article no. 1034. DOI: 10.3390/bioengineering11101034.

Надійшла до редакції 15.03.2025, розглянута на редколегії 20.05.2025

ЦИФРОВЕ ПРЕДСТАВЛЕННЯ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ АТОМНИХ ФУНКЦІЙ: ОСОБЛИВОСТІ ДЛЯ КОМП'ЮТЕРНОГО ЗОРУ ТА МАШИННОГО НАВЧАННЯ

В. О. Макаричев, В. В. Лукін,
С. С. Кривенко, І. В. Брисіна

Цифрові зображення, отримані за допомогою систем дистанційного зондування (ДЗ), стали важливими в численних технологічних застосуваннях у різних галузях, включаючи моніторинг навколишнього середовища, сільське господарство, міське планування та оборону. Порівняно з іншими типами даних, їхній значний розмір створює труднощі для ефективного застосування методів машинного навчання (МН) та комп'ютерного зору (КЗ). Зокрема, обробка таких великомасштабних даних може бути обчислювально ресурсоємною та трудомісткою, що ускладнює розгортання традиційних методів МН та КЗ у сценаріях, що вимагають реагування в режимі реального часу, або в системах з обмеженими ресурсами обробки, таких як автономні платформи. Одним з ключових питань у цьому контексті є розробка компактних представлень зображень, які зберігають важливі характеристики для подальшого аналізу. Ці представлення повинні зменшувати розмірність даних без втрати критичної інформації, необхідної для класифікації, кластеризації та інших завдань МН/КЗ. У цьому дослідженні ми досліджуємо дискретне атомарне перетворення (DAT), яке базується на атомарних функціях, як потенційне рішення цієї проблеми. Попередні дослідження показали, що DAT надає цінні переваги з точки зору стиснення та шифрування даних, забезпечуючи безпечне та ефективне зберігання та передачу. Метою цієї роботи є оцінка придатності DAT для застосувань машинного навчання (ML) та когерентної діагностики (CV), зокрема в контексті кластеризації зображень. Ми оцінюємо продуктивність відомого алгоритму кластеризації k-середніх при застосуванні до зображень, представлених за допомогою DAT. Експериментальні результати показують, що використання DAT значно скорочує час обчислення, досягаючи багатократного прискорення, без шкоди для якості кластеризації. Це свідчить про те, що DAT не тільки мінімізує розмір даних, але й зберігає структурні та статистичні особливості, важливі для завдань, заснованих на навчанні.

Ключові слова: представлення зображення; атомарна функція; дискретне атомне перетворення; атомні вбудовування; кластеризація зображень.

Макаричев Віктор Олександрович – канд. фіз.-мат. наук, доцент каф. інформаційно-комунікаційних технологій ім. О.О. Зеленського, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна.

Лукін Володимир Васильович – д-р техн. наук, проф., зав. каф. інформаційно-комунікаційних технологій ім. О. О. Зеленського, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна.

Кривенко Сергій Станіславович – канд. техн. наук, докторант каф. інформаційно-комунікаційних технологій ім. О.О. Зеленського, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна.

Брисіна Ірина Вікторівна – канд. фіз.-мат. наук, доц., доц. каф. вищої математики та системного аналізу, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна.

Viktor Makarichev – Candidate of Physics and Mathematics, Associate Professor at the Department of Information and Communication Technologies named after O. O. Zelensky, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine,
e-mail: v.makarichev@khai.edu, ORCID: 0000-0003-1481-9132.

Vladimir Lukin – Doctor of Technical Sciences, Professor, Head of the Department of Information and Communication Technologies named after O. O. Zelensky, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine,
e-mail: v.lukin@khai.edu, ORCID: 0000-0002-1443-9685.

Sergii Kryvenko – Candidate of Technical Science, Doctorate Student of the Department of Information and Communication Technologies named after O. O. Zelensky, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine,
e-mail: s.kryvevnko@khai.edu, ORCID: 0000-0001-6027-5442.

Iryna Brysina – Candidate of Physics and Mathematics, Associate Professor at the Department of Higher Mathematics and Systems Analysis, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine,
e-mail: i.brysina@khai.edu, ORCID: 0009-0009-4212-5964.