

# INTEGRATING FRACTAL DIMENSION AND TIME SERIES ANALYSIS FOR OPTIMIZED HYPERSPECTRAL IMAGE CLASSIFICATION

Khudayberdiev Mirzaakbar <sup>1</sup>, Tojiboyev Bobomurod <sup>2</sup>, Achilov Baxodir <sup>1</sup>,  
Alimkulov Nurmukhammad <sup>3</sup>, Ravshanov Anvar <sup>4</sup>

<sup>1</sup> Tashkent university of information technologies named after Muammad al-Khorazmi,  
Tashkent, Uzbekistan

<sup>2</sup> Andijan State University named after Zahiriddin Muhammad Babur,  
Andijan, Uzbekistan

<sup>3</sup> Kokand University Andijan Branch, Andijan, Uzbekistan

<sup>4</sup> Digital Technologies and Artificial Intelligence Research Institute,  
Tashkent, Uzbekistan

## ABSTRACT

*Hyperspectral images contain a huge number of spectral channels, which ensures high accuracy of analysis, but at the same time leads to problems associated with data redundancy, high computational load and decreased classification efficiency. In this paper, a method is proposed to integrate time series complexity analysis and fractal dimension (FD) to effectively reduce the dimensionality of hyperspectral data. Each pixel is considered as a time series characterized by spectral complexity, which allows identifying the most informative parts of the spectrum. Fractal dimension is used to quantify the complexity of spectral features and select significant channels. The proposed approach allows preserving critical information while minimizing losses during dimensionality reduction. To evaluate the effectiveness of the method, a comparison of the classification accuracy using the support vector machine (SVM) algorithm was carried out before and after applying the proposed optimization procedure. Experimental results on real hyperspectral data show significant dimensionality reduction while maintaining or improving classification quality.*

## KEYWORDS

*Permutation Entropy, hyperspectral data, Fractal Dimension, preferential entropy, The electromagnetic spectrum, Karusha-Kuna-Takker*

## 1. INTRODUCTION

Hyperspectral image is a dataset containing spatial and spectral characteristics of objects, making it an important tool in various fields such as agriculture, ecology, geology and remote sensing [1]. However, the high dimensionality and complexity of hyperspectral data create significant difficulties in their analysis and classification. Traditional methods of spectrum processing are often unable to fully take into account the dynamic and structural complexity of signals, which can reduce the accuracy of classification and interpretation of data [2].

To solve these problems, time series analysis methods are widely used to identify hidden patterns and features of signals. In particular, Permutation Entropy (PE) is a powerful tool for assessing the complexity of time series, reflecting the degree of order or chaos of spectral signals [3]. In

addition, the fractal dimension (FD) serves as a measure of structural complexity and self-similarity, which allows for a more profound characterization of the geometric properties of spectra[3]. In addition, the fractal dimension (FD) serves as a measure of structural complexity and self-similarity, which allows for a more profound characterization of the geometric properties of spectra [4]. This study aims to integrate time series complexity analysis using permutation entropy and fractal dimension estimation to optimize the classification process of hyperspectral data. The proposed approach allows improving feature extraction by taking into account both dynamic and structural aspects of spectra, which contributes to increasing the accuracy and reliability of classification. The paper examines the application of this technique on a real hyperspectral dataset and the effectiveness of the support vector machine (SVM) model [5].

## 2. MATERIALS AND METHODS

The purpose of this study is:

- 2.1. Description of hyperspectral data [6];
- 2.2. Time series complexity analysis [7];
- 2.3. Estimation of fractal dimension [8];
- 2.4. Feature selection algorithm [9];
- 2.5. Classification and evaluation (SVM) [10].

### 2.1. Description of Hyperspectral Data

Hyperspectral data are three-dimensional arrays consisting of two spatial dimensions (image width and height) and one spectral dimension (light intensity over a wide range of wavelengths). Each pixel in the image has its own spectrum, which can be used to analyze and identify various materials, vegetation types, and other objects.

*Hyperspectral imaging* is a data collection and analysis technology based on measuring electromagnetic radiation in a variety of narrow spectral bands extending across the entire visible and infrared spectrum (sometimes the ultraviolet spectrum) [11]. Unlike conventional photography, which records information in only a few wide bands (for example, the red, green, and blue channels), hyperspectral photography divides electromagnetic radiation into hundreds or even thousands of narrow spectral bands (Fig. 1) [12].

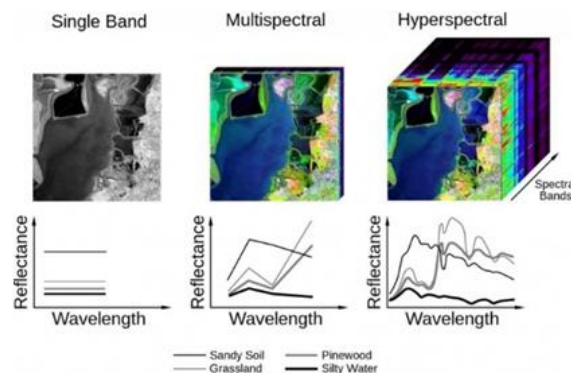


Fig. 1. Spectral bands

The electromagnetic spectrum describes all types of light, from very long radio waves, microwaves, infrared radiation, visible light, ultraviolet rays, and X-rays to very short gamma rays, most of which the human eye cannot see (Fig.) [13].

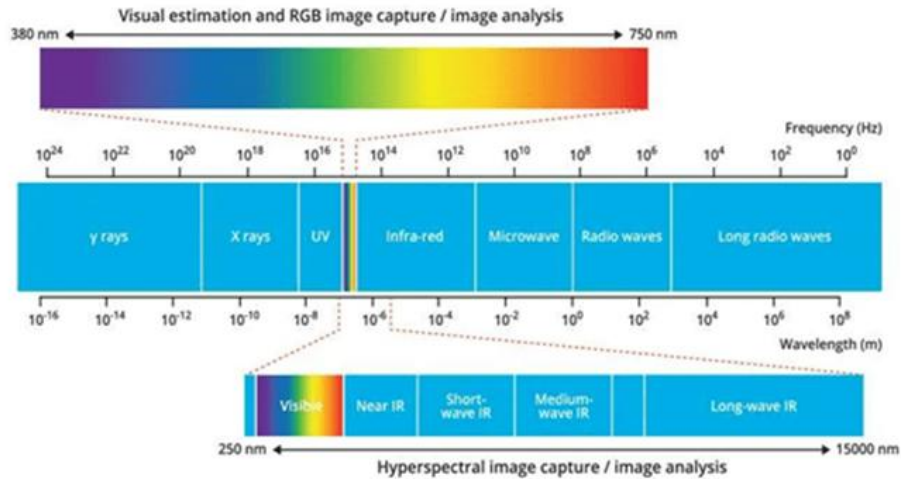


Fig. 2. Hyperspectral imaging captures wavelengths from 250 to 15,000 nm and thermal infrared radiation.

Hyperspectral images have high spectral but low spatial resolution, while multispectral images are characterized by high spatial but low spectral resolution. Data fusion studies have demonstrated that combining multi- and hyperspectral data allows for more accurate object classification. [14].

Hyperspectral sensors collect data as a set of images, each image in the set representing a narrowband range of wavelengths of the electromagnetic spectrum, also known as a spectral band. These images are combined to form a 3D hyperspectral data cube for processing and analysis. The hyperspectral cube contains spectral data in one dimension and spatial data in the other two, which can be used to create a detailed pixel-by-pixel chemical and spatial map (Fig 3) [15].

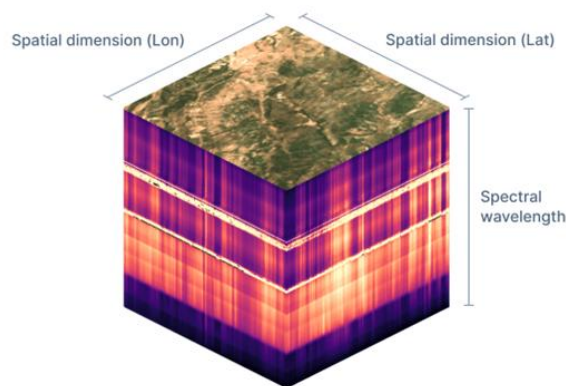


Fig 3. Hyperspectral data cube

The spatial and spectral characteristics of the obtained hyperspectral data are characterized by the information contained in its pixels [16]. Each pixel is a vector of values that define the intensities

at a particular location ( $x, y$  spatial coordinates) in different  $z$  ranges. The vector is known as the pixel spectrum, and it defines the spectral signature of the pixel located at  $(x, y)$ , i.e. the data stored in the pixel provides information about its spectrum over the entire range of the sensor used [17]. Pixel spectra are important characteristics in the analysis of hyperspectral data. But these pixel spectra are distorted by a number of factors (sensor noise, atmospheric effects, low resolution, etc.) (Fig. 4).

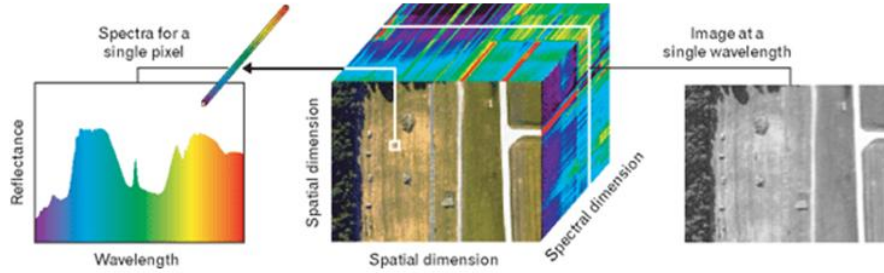


Fig. 4. Pixel space of hyperspectral data

Satellite images obtained using hyperspectral sensors are not as widely available as multispectral ones, due to the small number of spacecrafts with appropriate sensors on board and the high cost of the images obtained. [18].

Advantages of hyperspectral data:

1. More spectral bands.
2. Better object discrimination ability
3. More accurate chemical composition analysis
4. Wider scope of use

Despite its advantages, hyperspectral data also has a number of disadvantages:

1. High demands on computing power
2. Expensive equipment
3. Limited spatial resolution of images
4. Expertise is required to interpret the data.

Hyperspectral images containing  $N$  pixels and  $B$  spectral channels (bands) were used as the initial data. Each pixel  $x_i$  is represented as a vector of spectral values [19]:

$$x_i = [x_{i1}, x_{i2}, \dots, x_{iB}], \in \mathbb{R}^B, i = 1, 2, \dots, N \quad (1)$$

Thus, the hyperspectral image can be represented as a matrix  $X \in \mathbb{R}^{N \times B}$ .

## 2.2. Time Series Complexity Analysis

Time series analysis is a set of mathematical and statistical methods of analysis designed to identify the structure of time series and to forecast them [20]. This includes, in particular, regression analysis methods. Identifying the structure of a time series is necessary in order to construct a mathematical model of the phenomenon that is the source of the analyzed time series. Forecasting future values of a time series is used for effective decision making [21].

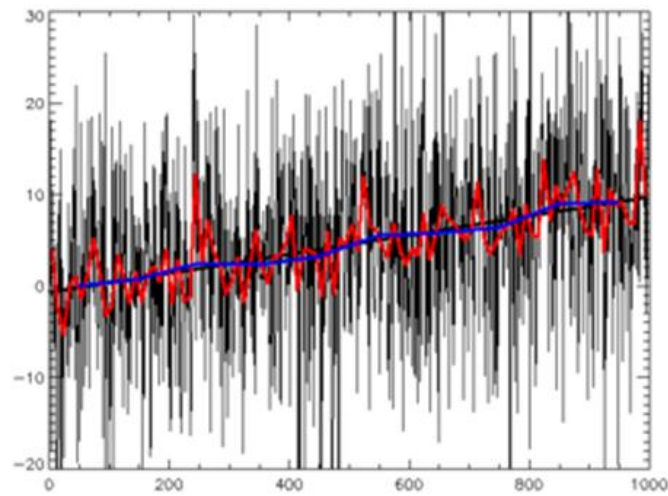


Fig. 5. Example of a time series

Time series consist of two elements:

- the time period for which or as of which numerical values are given;
- numerical values of a particular indicator, called series levels.

Time series are classified according to the following criteria:

- according to the form of the level's representation;
- the series of absolute indicators;
- relative indicators;
- average values.
- by the number of indicators for which levels are determined at each point in time: one-dimensional and multidimensional time series;
- by the nature of the time parameter: instantaneous and interval time series. In instantaneous time series, levels characterize the values of the indicator as of certain points in time. In interval series, levels characterize the value of the indicator for certain periods of time. An important feature of interval time series of absolute values is the possibility of summing their levels. Individual levels of the moment series of absolute values contain elements of repeated counting. This makes summing up the levels of moment series meaningless;
- according to the distance between dates and time intervals, equidistant ones are distinguished — when the registration dates or the end of periods follow each other at equal intervals, and incomplete (unequally spaced) ones — when the principle of equal intervals is not respected;
- by missing values: full and incomplete time series;
- time series can be deterministic and random: the former are obtained based on the values of some random function (a series of consecutive data on the number of days in months); the latter is the result of the realization of some random variable;
- depending on the presence of the main trend, stationary series are distinguished, in which the average value and variance are constant, and non-stationary ones containing the main trend of development.

Each pixel is considered as a time series consisting of intensity values in different spectral bands. To estimate the complexity of the time series, the permutation Entropy (PE) proposed by Band and Pompe is used. [22]:

$$PE(x_i) = \sum_{j=1}^{n!} p_i \log p_j \quad (2)$$

where:

- $p_j$  is the probability of occurrence of each of  $n!$  possible permutations of length  $n$  in the series in the series  $x_i$ ;
- $n$  is the order of permutation  $n \in \mathbb{Z}$ .

This indicator is used to identify areas of the spectrum that contain a complex and, therefore, informative structure.

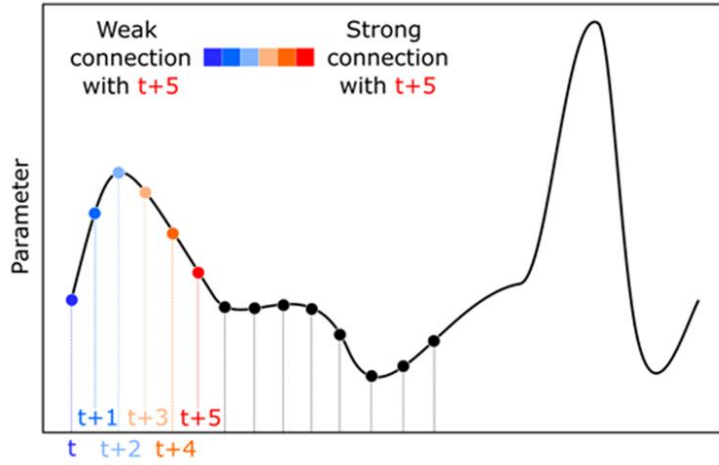


Fig 6. Parameters of the informative structure

### 2.3. Estimation of Fractal Dimension

Fractal dimension is one of the ways to determine the size of a set. In metric space, it is called the fractal dimension of an  $n$ -dimensional set and has several varieties. They are calculated as follows [23].

*Fractal (Hausdorff, box-counting) dimensions are calculated using the formula*

$$d = \lim_{\varepsilon \rightarrow 0} \frac{\ln N(\varepsilon)}{\ln \frac{1}{\varepsilon}} \quad (3)$$

where,  $N(\varepsilon)$  is the minimum number of cubes with side  $\varepsilon$ , required to cover the entire complex. The measurement is defined as an exponent of the degree of  $d$  in  $N(\varepsilon) \propto \frac{1}{\varepsilon^d}$  (Fig. 7).

(a) Dividing blocks by a boundary line

(b) Division of total volume blocks



Fig 7. Hausdorff, box-counting dimension

- a) Border block selection:  $N(\varepsilon) \propto \frac{1}{\varepsilon}, d = 1,$
- b) Division of total volume blocks:  $N(\varepsilon) \propto \frac{1}{\varepsilon^2}, d = 2.$

*Another fractal dimension method is the shoreline method*

The length of the coastline is measured in  $l$ , then the measured length is calculated using the formula (4):

$$L = \Delta l^{-\alpha}, \Delta = \text{const} \quad (4)$$

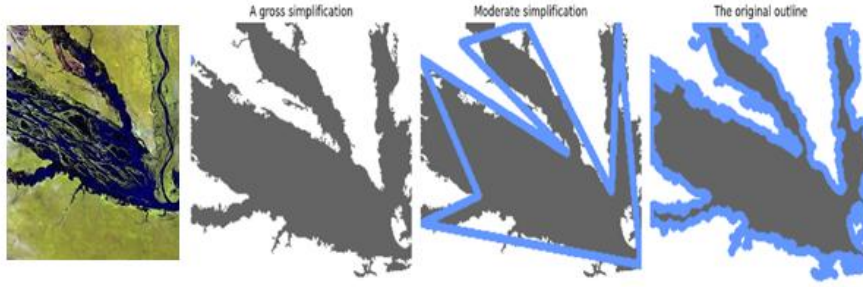


Fig 8. The total length of the coastline

Fig. 8, demonstrating traditional ideas about geometry, forms a scale in accordance with predictable, understandable and familiar ideas about the space in which they are located. For example, take a line, divide it into three equal parts, and then each part will be three times smaller than the length of the original line. The same thing happens on the plane. If you measure the area of a square, and then measure the area of a square by  $\frac{1}{3}$  the length of the side of the original square, it will be 9 times smaller than the area of the original square [24]. This measurement can be determined mathematically using the rule of measurement according to the formula (5):

$$N \propto \varepsilon^{-D} \quad (5)$$

where,  $N$  is the number of parts,  $\varepsilon$  is the dimensional coefficient,  $N$  is the dimensional coefficient,  $\propto$  is the fractal dimension, means the proportion in this sign. This scaling rule confirms the traditional scaling rules of geometry, since for a line  $N = 3$ , when  $\varepsilon = \frac{1}{3}$ , then  $D = 1$ , and for squares, because  $N = 9$ , when  $\varepsilon = \frac{1}{3}$ ,  $D = 2$ . The same rule applies to fractal geometry, but it is less intuitive. To calculate the unit length of a fractal line, at first glance, reduce the scale by three times, in this case  $N = 3$ , when  $\varepsilon = \frac{1}{3}$  and we get the value of formula (5) by changing formula (6):



$$\log_{\varepsilon} N = -D \frac{\log N}{\log \varepsilon} \quad (6)$$

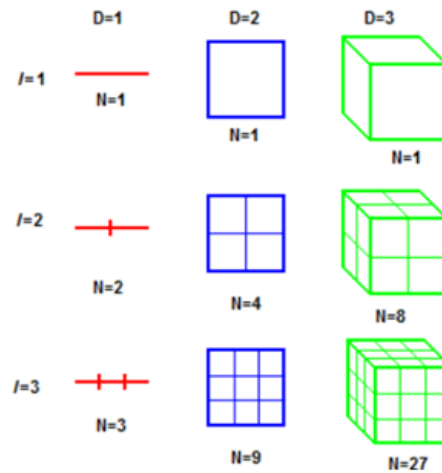


Fig. 9. Traditional representation of geometry in measurements and scale definition

## 2.4. Feature Selection Algorithm

Feature selection algorithms are methods used in machine learning and data mining to select the most relevant features from a set of input data [25]. This is done to optimize the model, reduce complexity and improve performance.

Main categories of feature selection algorithms:

### Filters:

- Assess the importance of each feature independently of the model using metrics such as correlation coefficient, mutual information, Fisher criterion and others;
- Examples: Pearson Correlation Coefficient, Feature Importance Assessment with LightGBM;
- Advantages: Fast, independent of model type;
- Disadvantages: May not take into account interactions between features.

### Wrappers:

- Estimate the importance of features based on the performance of a model trained on different subsets of features;
- Examples: Sequential Feature Elimination, Direct Feature Selection (Forward Selection);
- Advantages: Takes into account the relationships between features;
- Disadvantages: Computationally expensive.

### Embedded Methods:

- Feature selection occurs as part of the model training process, for example when using regularization;
- Examples: L1 regularization (Lasso), L2 regularization;
- Advantages: Computationally less expensive than wrappers;
- Disadvantages: May be specific to the model type.



### Hybrid methods:

- Combines approaches of filters, wrappers and built-in methods.

### Reducing overfitting:

- Feature selection can help avoid overfitting the model on data specific to the current dataset.
- Examples of feature selection algorithms [26].

### mRMR (Minimal Redundancy Maximum Relevance):

- Maximizes the mutual information between features and the target variable while minimizing the mutual information between the selected features.

### Evolutionary selection of traits (EFS):

- Using evolutionary algorithms to find the optimal subset of features.

### Recursive Feature Elimination (RFE):

- Iterative feature removal based on model-estimated importance.

### Forward Selection: Step-by-step addition of features that improve the quality of the model [27].

- The proposed method combines complexity analysis and fractal dimension;
- For each pixel  $x_i$  the permutation entropy  $PE(x_i)$  is calculated;
- Spectral channels with the highest  $PE$  values are aggregated;
- For each  $k$  channel, the  $D_k$  fractal dimension is calculated;
- The channels are rated by  $D_k$ , and the  $M$  channels with the highest values are selected;
- A reduced matrix of features  $X' \in \mathbb{R}^{N \times M}$  is formed, where  $M \ll B$ .

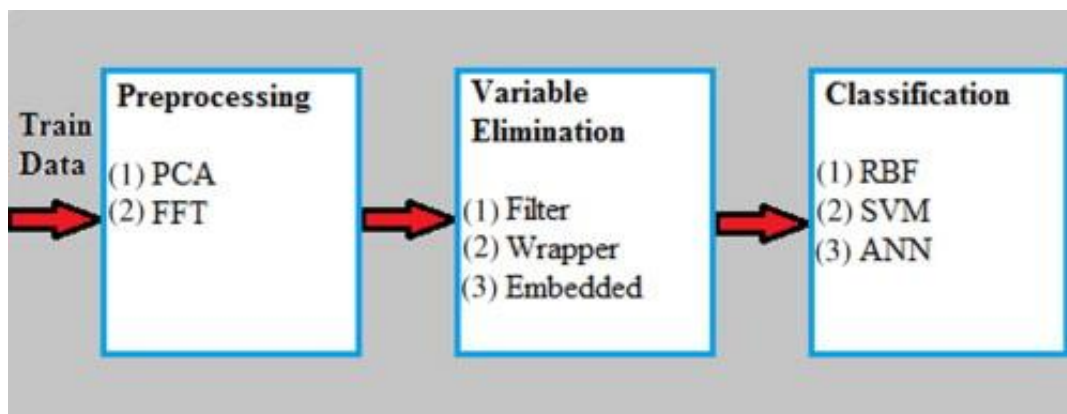


Fig 10. Feature selection

## 2.5. Classification and Evaluation (Support Vector Machine)

Support Vector Machine (SVM) is a machine learning algorithm that is used for classification and regression tasks. The main idea of SVM is to find the optimal separating hyperplane in a multidimensional space that separates objects of different classes as much as possible [28].

In the case of a classification problem, SVM seeks to find a hyperplane such that the distance from it to the nearest points of the training sample (support vectors) is maximal. These support vectors are the data points that are closest to the hyperplane and play a key role in determining the position of the hyperplane.

The main SVM methodologies include the following:

- SVM for the Linearly Inseparable Case;
- SVM with nonlinear kernels
- Probabilistic assessment of the quality of models;

*SVM for the Linearly Inseparable Case* [29].

Set the simplest linear model of the form:

$$a(x) = \text{sign}(\omega^T x - b) = \text{sign}((\omega, x) - b) = \text{sign}(\omega * x - b) \quad (7)$$

There are three ways, but all of them mean a linear combination of the parameter vector  $\omega$  with the image  $x$  and adding a  $-b$  offset. At the output, the model produces values,  $a(x) \in \{-1; +1\}$ . Next, assume that the training sample consists of linearly separable images (then, this case generalizes to linearly inseparable). Then the width of the strip will be determined by the location of the boundary vectors  $x$  in the feature space. In general, images in training samples are rarely linearly separable.

Therefore, with a linearly inseparable sample, it will not be possible to find parameters  $\omega$  and  $b$ , that would satisfy the linear constraints on the offsets:

$$M_i(\omega, b) \geq 1, \quad i = 1, 2, \dots, l \quad (8)$$

When allowing the classifier to make a mistake by a certain value (slack variables),  $\xi_i \geq 0$ ,  $i = 1, 2, \dots, l$  for each  $i$ -th image,  $M_i(\omega, b) \geq 1 - \xi_i$ , where  $i = 1, 2, \dots, l$ , the values of slack variables can still be perceived as a penalty for violating the original inequality. If all  $\xi_i \rightarrow +\infty$ , then any weights can be taken, for example 0, and then the optimization problem will be solved. Here it is allowed to make mistakes, but the magnitude of this error should be as small as possible, that is, it is necessary to find such  $\omega$  and  $b$ , that  $\xi_i \rightarrow 0$ , where  $i = 1, 2, \dots, l$ , this condition can be taken into account in the minimization algorithm, writing it as follows:

$$\begin{cases} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^l \xi_i \xrightarrow{\omega, b, \xi} \min \\ M_i(\omega, b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, l \\ \xi_i \geq 0, \quad i = 1, 2, \dots, l \end{cases} \quad (9)$$

where,  $C$  is a hyperparameter that determines the degree of minimization of values  $\{\xi_i\}$ .

All the obtained equivalent optimization problems are considered for the general case of a linearly inseparable sample. Now here we can rewrite the last two inequalities in the system as:

$$\begin{cases} \xi_i \geq 1 - M_i(\omega, b), & i = 1, 2, \dots, l \\ \xi_i \geq 0, & i = 1, 2, \dots, l \end{cases} \quad (10)$$

And, since the solution to the problem of finding the minimum of the coefficients  $\omega$  and  $b$  has been obtained, then in this inequality it is logical to choose equality, that is:

$$\begin{cases} \xi_i = 1 - M_i(\omega, b) \\ \xi_i = 0 \end{cases} \Rightarrow L_i(\omega, b) = \max(0, 1 - M_i(\omega, b)), \quad i = 1, 2, \dots, l \quad (11)$$

This expression is also written like this:

$$L_i(\omega, b) = (1 - M_i(\omega, b))_+, \quad i = 1, 2, \dots, l \quad (12)$$

As a result, the original system becomes equivalent to an unconstrained minimization problem:

$$C \sum_{i=1}^l (1 - M_i(\omega, b))_+ + \frac{1}{2} \|\omega\|^2 \xrightarrow{w, b} \min \quad (13)$$

or, when divided by parameter  $C$ , available:

$$\sum_{i=1}^l (1 - M_i(\omega, b))_+ + \frac{1}{2C} \|\omega\|^2 \xrightarrow{\omega, b} \min \quad (14)$$

The optimization problem for the support vector method was obtained using system (9), and from it we arrived at an algorithm for finding the parameters  $\omega$  and  $b$  using formula (14). Conventional gradient methods are not suitable for minimizing this functional here, since the loss function here is continuous but not smooth (derivatives do not exist at the inflection point)[30]. As an option, you can use subgradient methods, that is, calculate the derivative according to the rule:

$$\nabla J(\omega, b) = \nabla(1 - M_i(\omega, b)) = \begin{cases} -1, & J(\omega, b) < 1, \\ 0, & J(\omega, b) \geq 1 \end{cases} \quad (15)$$

But initially, the solution to the problem of optimization of the support vector method was reduced to solving system (9). That is, the problem of quadratic programming, minimization of the coefficients  $\omega$  under linear constraints in the form of inequalities. This approach leads to fairly effective numerical methods and in addition, allows us to select objects (observations) on the basis of which the coefficients  $\omega$  and  $b$  are calculated. This is interesting additional information about the structure of the training set [31]. As a result, we come to the conclusion that the coefficients  $\omega$  are calculated using the following formula:

$$\omega = \sum_{i=1}^h \lambda_i y_i x_i \quad (16)$$

where,  $\lambda_i$  are some coefficients that are also calculated during the solution of this optimization problem.

*SVM with nonlinear kernels.* The coefficients  $\omega$  of the linear binary classifier in formula (1), in the support vector machine will be calculated using formula (10), where  $\{\lambda_i\}$  are some coefficients [32]. Moreover, non-zero coefficients  $\{\lambda_i\} \neq 0$  correspond to support or error vectors

(outliers). If we combine formulas (1) and (10), we get the general formula (17) of the linear classifier model:

$$\omega = \text{sign}(\sum_{i=1}^h \lambda_i y_i (x_i, x) - b) \quad (17)$$

here, the sum is taken not for all objects, but only for reference ones, for which  $\{\lambda_i\} \neq 0$ .

The classifier computes the weighted sum of the scalar products of the support vectors  $\{x_i\}_{i=1}^h$  with some input vector  $x$ , subtracts the bias  $b$ , and determines the sign. Graphically it looks like this:

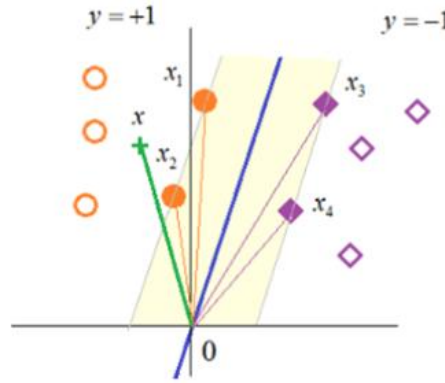


Fig 11. The classifier calculates the weighted sum of the scalar products of the support vectors

Let  $x_1, x_2, x_3, x_4$  be a linearly separable sample with four boundary vectors. For simplicity, the bias will be set to zero ( $b = 0$ ). Then for an arbitrary vector  $x$  the following linear combination will be calculated:

$$\lambda_1(x_1, x) + \lambda_2(x_2, x) - \lambda_3(x_3, x) - \lambda_4(x_4, x) \quad (18)$$

Or, it can be written in this form. First, the support vectors for the first and second classes of images are summed:

$$\begin{cases} \omega_+ = \lambda_1 x_1 + \lambda_2 x_2 \\ \omega_- = \lambda_3 x_3 + \lambda_4 x_4 \end{cases} \quad (19)$$

then, the classifier assigns a sign to the projection of the vector  $x, (\omega_+, x) - (\omega_-, x)$ , and produces its decision:

$$a(x) = \text{sign}((\omega_+, x) - (\omega_-, x)) \quad (20)$$

Formula (20) interprets the work of a linear binary classifier in the support vector machine.

*SVM with nonlinear kernels.* Formula (17) of the linear classifier model is reduced to the following form:

$$(x_i, x) = (x_i^T, x), \quad i = 1, 2, \dots, h \quad (21)$$

Formula (21) is squared:

$$(x_i, x)^2 = (x_i^T x)^2, \quad i = 1, 2, \dots, h \quad (22)$$

For the specified transformation and a number of other nonlinear transformations, the solution of the system by the *Karush-Kuhn-Tucker* method using the Lagrange function remains unchanged [33].

Now consider a function  $K(x, x')$ , which is a kernel (suitable for the SVM problem) if and only if it is symmetric:  $K(x, x') = K(x', x)$  and non-negative definite:

$$\iint K(x, x')g(x)g(x')dxdx' \geq 0 \quad (23)$$

for any  $g: X \rightarrow \mathbb{R}$ .

This function  $K(x, x') = K(x', x)$  and it will be written for two-dimensional vectors:  $u = [u_1, u_2]^T, v = [v_1, v_2]^T$ , from here it will turn out  $K(u, v) = (u, v)^2 = (u_1v_1 + u_2v_2)^2 = u_1^2v_1^2 + 2u_1v_1u_2v_2 + u_2^2v_2^2 = \left([u_1^2, u_2^2, \sqrt{2}u_1u_2]\right)^T \cdot \left([v_1^2, v_2^2, \sqrt{2}v_1v_2]\right)^T$ . That is, squaring the scalar product of two-dimensional vectors is analogous to the scalar product of three-dimensional vectors [34, 35]:

$$\begin{cases} \Psi(u) = \left([u_1^2, u_2^2, \sqrt{2}u_1u_2]\right)^T \\ \Psi(v) = \left([v_1^2, v_2^2, \sqrt{2}v_1v_2]\right)^T \end{cases} \quad (24)$$

The function  $\Psi(x)$  forms a new three-dimensional feature space in which the usual linear support vector machine algorithm begins to operate. But from the position of the original two-dimensional feature space, we obtain polynomial functions of level two [36]. Moreover, the polynomials here are not arbitrary, but consist only of homonyms of degree two (there are no terms of lower degrees here). Separating hyperplanes for different types of kernels:

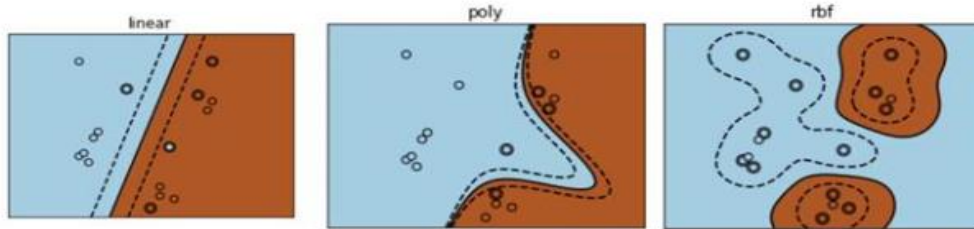


Fig. 12. Separating hyperplanes for different types of kernels (liner, poly, rbf).

Here, linear is the usual scalar product, poly is the polynomial kernel formed by functional transformations of the form:

$$\begin{cases} K(x, x') = (x, x')^d \\ K(x, x') = ((x, x') + 1)^d \end{cases} \quad (25)$$

rbf – radial cores defined by the expression:

$$K(x, x') = \exp(-\lambda||x - x'||^2) \quad (26)$$

core of the species:

$$K(x, x') = th(k_1(x, x') - k_0), \quad k_1, k_1 \geq 0 \quad (27)$$

On its basis, an analogue of a two-layer neural network with sigmoid activation functions is obtained.

*SVM as a two-layer neural network.* SVM can be represented as the following computational structure on a two-layer neural network [37]. Since the output of the model for arbitrary kernels is calculated using formula (11), a two-layer neural network is formed.

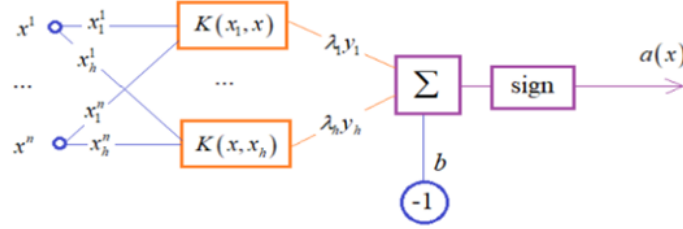


Fig. 13. Two-layer neural network on the SVM model

On the hidden layer, the convolutions of the input vector  $x$  with the support vectors  $x_1, \dots, x_h$  are calculated taking into account the selected kernel  $K(x, x')$ . Then, all these values are multiplied by the weighting coefficients  $\lambda_1 y_1, \dots, \lambda_h y_h$ , summed up and passed through the signed activation function. Moreover, SVM immediately determines the required number of neurons in the hidden layer and, of course, the values of the weight coefficients.

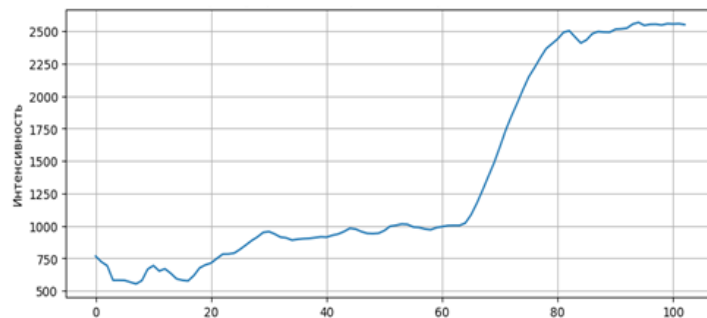
*Methods of kernel synthesis.* In conclusion, you can learn simple rules for synthesizing kernels for the support vector machine. The main approaches include the following:

- $K(x, x') = (x, x')$  – the scalar multiplications;
- $K(x, x') = 1$  – constant;
- $K(x, x') = K_1(x, x')K_2(x, x')$  – multiplication of kernels (suitable for SVM);
- $K(x, x') = (\Psi(x), \Psi(x')), \forall \Psi: X \rightarrow \mathbb{R}$  – function application;
- $K(x, x') = \alpha_1 K_1(x, x') + \alpha_2 K_2(x, x'), \quad \alpha_1, \alpha_2 > 0$  – the sum of the cores.

### 3. RESULT

The results of the application of the proposed method for reducing the dimension of hyperspectral data based on the analysis of the complexity of time series and fractal dimension (FD). All experiments were conducted using a hyperspectral dataset from the platform. <https://www.kaggle.com/>, containing  $N = 31344$  pixels and  $B = 105$  spectral channels. For each pixel of the image  $x_i \in \mathbb{R}^B$  the entropy of permutations was calculated using the formula (2)

a) Time series in pixels (spectrum)



b) Preferential entropy ( $PE$ )

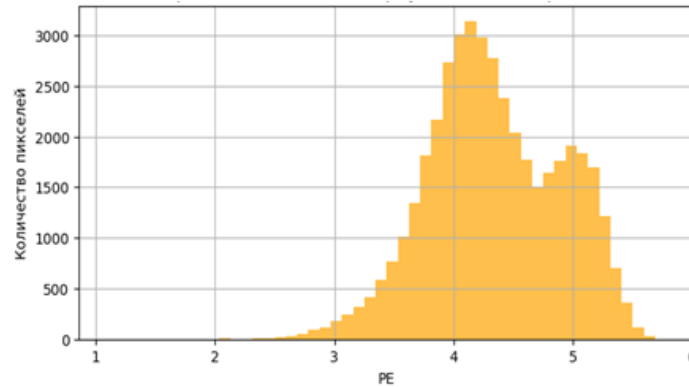


Fig14. a) time series in pixels (spectrum), б) preferential entropy ( $PE$ )

- High  $PE$  values correspond to spectral values with a complex and informative structure.
  - Spectral ranges with the highest average entropy across all pixels were identified.
- The result: about 35-40 spectral channels showed high complexity and were selected for the next stage.

For each selected spectral channel  $k$ , the fractal dimension was calculated using the formula (3,4,5,6).

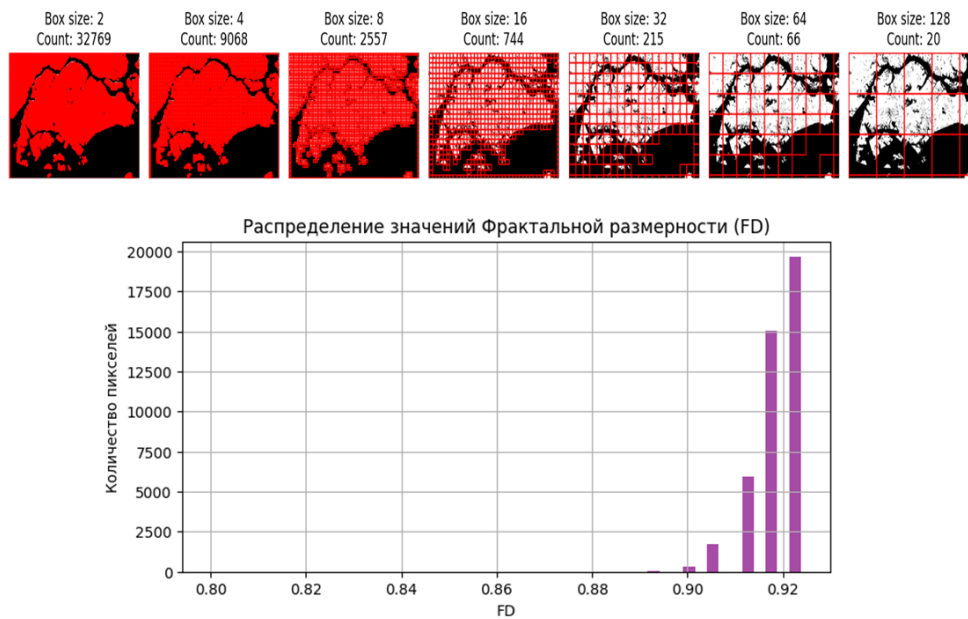


Fig 15. Fractal dimension

Linear regression on a logarithmic scale was used for  $\varepsilon \in [0.90, 0.93]$ . Result: 35-40 channels with the highest  $D_k$ , values corresponding to the highest structural complexity were selected.

The classification was used SVM with an RBF core. The assessment was based on metrics:

- Accuracy, F1-score, Precision, Recall.



Table 1

SVM Classification Report				
	precision	recall	f1-score	support
1	0.91	0.93	0.92	321
2	0.81	0.83	0.82	381
3	0.91	0.83	0.87	326
4	0.93	0.84	0.89	400
5	0.88	0.99	0.93	373
6	0.86	0.89	0.87	360
7	0.85	0.81	0.83	374
8	0.96	0.90	0.93	367
9	0.98	0.99	0.99	497
10	0.92	0.99	0.95	398
<b>accuracy</b>			0.90	3797
<b>macroavg</b>			0.90	3797
<b>weighted avg</b>			0.90	3797

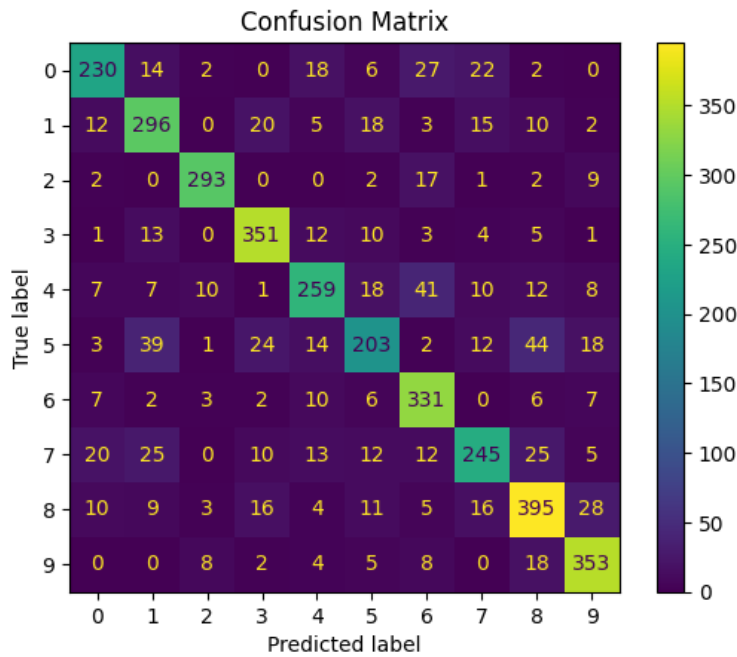


Fig 16. Confusion matrix

The proposed method showed a better classification quality compared to PCA and complete data. Reducing the dimension by more than 90% has allowed for faster learning and lower computing costs. Using the GPU gave an almost 10-fold acceleration of calculations compared to the CPU.

#### 4. ANALYSIS

The conducted analysis confirmed the effectiveness of the combined approach using permutation entropy (PE) and fractal dimension (FD) for the selection of informative features in hyperspectral

data. PE allowed us to identify pixels with high spectral complexity, which contributes to improved classification. FD accurately identified spectral channels with a pronounced texture structure, providing a more informed choice of features. The combined use of these metrics allowed us to reduce the dimensionality by more than 90% without losing classification quality. The method showed an advantage over traditional approaches, preserving the physical interpretability of features. Despite the high computational load, the use of GPU provided acceptable processing time.

## 5. DISCUSSION

The proposed method of integrating permutation entropy and fractal dimension analysis has shown high efficiency in classifying hyperspectral data. Unlike traditional dimensionality reduction (e.g. PCA), this approach preserves the physical meaning of spectral features and provides improved interpretability. Experiments confirmed that PE and FD-based features can improve the classification accuracy of SVM while reducing the data dimensionality by 80–90%. Despite the computational cost, the GPU implementation provides high performance, making the method suitable for practical application.

## 6. CONCLUSION

In this paper, a new method for dimensionality reduction of hyperspectral data based on the integration of time series complexity analysis and fractal dimension estimation was developed and investigated. The proposed approach allows for the efficient identification of the most informative spectral channels, taking into account both the local complexity of each pixel and the global structural characteristics of the spectrum.

Experimental results on real hyperspectral data have shown that the method significantly reduces the dimensionality of the original data (up to 90%) without significant loss, and often with improved classification accuracy compared to classical methods such as PCA. The use of parallel computing on GPUs allowed for faster processing, making the method applicable to real-time tasks.

Thus, the integration of complexity analysis and fractal dimension represents a promising tool for optimizing hyperspectral images and improving the efficiency of their subsequent analysis and classification. In the future, it is planned to expand the method using other complexity measures and deep learning for even more accurate feature selection.

## REFERENCE

- [1] Jiale Zhao, Dan Fang, Jiaju Ying, Yudan Chen, Qi Chen, Qianghui Wang, Guanglong Wang, Bing Zhou, "A camouflage target classification method based on spectral difference enhancement and pixel-pair features in land-based hyperspectral images," *Engineering Applications of Artificial Intelligence*, vol. Volume 156, no. ISSN 0952-1976, <https://doi.org/10.1016/j.engappai.2025.111141>, p. Part A, 2025.
- [2] L. Pichon, Q. Lemasson, D. Bachiller-Perea, C. Pacheco, "Advances in dynamic and batch processing of PIXE spectra," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. Volume 565, no. ISSN 0168-583X, <https://doi.org/10.1016/j.nimb.2025.165734>, 2025.
- [3] Dongfeng Yang, Jun Hu, "Fast identification of maize varieties with small samples using near-infrared spectral feature selection and improved stacked sparse autoencoder deep learning," *Expert Systems with Applications*, vol. <https://doi.org/10.1016/j.eswa.2025.128265>, pp. ISSN 0957-4174, 2025.

- [4] Javier Del-Pozo-Velázquez, Javier Manuel Aguiar-Pérez, Pedro Chamorro-Posada, María Ángeles Pérez-Juárez, Xinheng Wang, Pablo Casasaca-de-la-Higuera, "Smoke detection in images through fractal dimension-based binary classification," *Digital Signal Processing*, vol. Volume 166, no. ISSN 1051-2004, <https://doi.org/10.1016/j.dsp.2025.105346>, 2025.
- [5] Arun Kumar, Nishant Gaur, Aziz Nanthaamornphong, "A mathematical PAPR estimation of OTFS network using a machine learning SVM algorithm," *Results in Optics*, vol. Volume 21, no. ISSN 2666-9501, <https://doi.org/10.1016/j.rio.2025.100834>, 2025.
- [6] Bingjie Lu, Yinyin Zhang, Zhangyun Gao, Yongqi Chen, Shen Su, Xiao Hu, Jing Guo, Wanneng Yang, Hui Feng, "DEA: Hyperspectral data high-throughput extraction and analysis software," *Smart Agricultural Technology*, vol. Volume 10, no. ISSN 2772-3755, <https://doi.org/10.1016/j.atech.2025.100800>, 2025.
- [7] Iván Gallo-Méndez, Jaime Clark, Denisse Pastén, "Time series analysis of wildfire propagation in Chile: A complex networks approach," *Chaos, Solitons & Fractals*, vol. Volume 196, no. ISSN 0960-0779, <https://doi.org/10.1016/j.chaos.2025.116347>, 2025.
- [8] Mizaakbar Khudaiberdiev, Igor Khan, Bobomurod Tojiboyev, Bahodir Achilov, "Fractal representations in image processing of remote sensing of the earth," *E3S Web of Conferences*, vol. Volume 541, no. ISSN: 2117-4458, <https://doi.org/10.1051/e3sconf/202454104010>, 2024.
- [9] Shaoming Qiu, Jingjie He, Yan Wang, Bicong E, "A Feature Selection Method for Software Defect Prediction Based on Improved Beluga Whale Optimization Algorithm," *Computers, Materials and Continua*, vol. Volume 83, no. Issue 3, pp. Pages 4879-4898, 2025.
- [10] M.Khodayberdiev, N.Alimkulov, B.Achilov, "Classification of Lung Cancer Diseases by Support Vector Method (In Russian)," *Science and Innovative Development*, vol. 7(1), no. doi:10.36522/2181-9637-2024-1-1, pp. 8-20, 2024.
- [11] Jianan Zhang, Qing Zhang, Jiansheng Wang, Yan Wang, Qingli Li, "A dual branch based stitching method for whole slide hyperspectral pathological imaging," *Displays*, vol. Volume 89, no. ISSN 0141-9382, <https://doi.org/10.1016/j.displa.2025.103090>, 2025.
- [12] N. Anand, M. Balasingh Moses, "Electromagnetic radiation detection and monitoring in high-voltage transmission lines using machine learning techniques," *Measurement*, vol. Volume 253, no. ISSN 0263-2241, <https://doi.org/10.1016/j.measurement.2025.117645>, p. Part C, 2025.
- [13] Ebrahim Gholami Hatam, "A 3-D simulation model for X-ray path length and yield variation in micro-PIXE analysis of rough surfaces," *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. <https://doi.org/10.1016/j.sab.2025.107238>, pp. ISSN 0584-8547, 2025.
- [14] Yongxian Wang, Mingchao Shao, Jiacheng Wang, Jingwei An, Jianshuang Wu, Xia Yao, Xiaohu Zhang, Chongya Jiang, Tao Cheng, Yongchao Tian, Weixing Cao, Dong Zhou, Yan Zhu, "A novel snapshot multispectral imaging sensor for quantitative monitoring of crop growth," *Plant Phenomics*, vol. <https://doi.org/10.1016/j.plaphe.2025.100056>, pp. ISSN 2643-6515, 2025.
- [15] Ajanta Goswami, Bikram Banerjee, Bharat Joshi, Abhishek Kumar, Hrishikesh Kumar, Angana Saikia, "A statistical technique for noise identification and restoration of hyperspectral image cube," *Remote Sensing Applications: Society and Environment*, vol. Volume 27, no. ISSN 2352-9385, <https://doi.org/10.1016/j.rsase.2022.100794>, 2022.
- [16] Wenfeng Yang, Xin Zheng, Ziran Qian, Shaolong Li, Yu Cao, Guo Li, Yikai Yang, Yue Hu, Shuangqi Lyu, Zihao Li, Wenxuan Wang, "Influence of surface morphology on the spectral characteristics of LIBS for laser paint removal of aircraft skins," *Talanta*, vol. Volume 293, no. ISSN 0039-9140, <https://doi.org/10.1016/j.talanta.2025.128097>, 2025.
- [17] Wei Yang, Wenwen Meng, Dongfeng Shi, Linbin Zha, Yafeng Chen, Jian Huang, Yingjian Wang, "Single-pixel edge imaging with gradient Radon spectrum," *Optics Communications*, vol. Volume 529, no. ISSN 0030-4018, <https://doi.org/10.1016/j.optcom.2022.129064>, 2025.
- [18] Xudayberdiyev M.X., Tojiboyev B.M., F.K.Samadova, "ALGORITHMS FOR FEATURE EXTRACTION AND OPTIMISATION OF OBJECT RECOGNITION OPERATOR," *CHEMICAL TECHNOLOGY. CONTROL AND MANAGEMENT*, pp. №2 (122) pp.46-54, 2025.
- [19] Miriam Medina-García, José M. Amigo, Miguel A. Martínez-Domingo, Eva M. Valero, Ana M. Jiménez-Carvelo, "Strategies for analysing hyperspectral imaging data for food quality and safety issues – A critical review of the last 5 years," *Microchemical Journal*, vol. Volume 214, no. ISSN 0026-265X, <https://doi.org/10.1016/j.microc.2025.113994>, 2025.

- [20] Mohamed Abuella, Hadi Fanaee, Slawomir Nowaczyk, Simon Johansson, Ethan Faghani, "Time-series analysis approach for improving energy efficiency of fixed-route passenger vessel in short-sea shipping," *Ocean Engineering*, vol. Volume 334, no. ISSN 0029-8018, <https://doi.org/10.1016/j.oceaneng.2025.121555>, 2025.
- [21] Md Mohasin Howlader, Md Mazharul Haque, "Opposing-through crash risk forecasting using artificial intelligence-based video analytics for real-time application: integrating generalized extreme value theory and time series forecasting models," *Accident Analysis & Prevention*, vol. Volume 218, no. ISSN 0001-4575, <https://doi.org/10.1016/j.aap.2025.108073>, 2025.
- [22] Xirui Li, Zhen Li, Yong Deng, "An improved random permutation set entropy to address self-contradiction," *Expert Systems with Applications*, vol. Volume 288, no. ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2025.128218>, 2025.
- [23] Christos Antrias, Alexios Ioannidis, Thomas Tsovilis, "Fractal dimension analysis of lightning discharges of various types based on a comprehensive literature review," *Atmospheric Research*, vol. Volume 312, no. ISSN 0169-8095, <https://doi.org/10.1016/j.atmosres.2024.107736>, 2024.
- [24] Rami Ahmad El-Nabulsi, "A model for ice sheets and glaciers in fractal dimensions," *Polar Science*, Vols. ISSN 1873-9652, , no. <https://doi.org/10.1016/j.polar.2025.101171>, 2025.
- [25] Chuili Chen, Xiangjuan Yao, Dunwei Gong, Huijie Tu, "A multi-objective evolutionary algorithm for feature selection incorporating dominance-based initialization and duplication analysis," *Swarm and Evolutionary Computation*, vol. Volume 95, no. ISSN 2210-6502, <https://doi.org/10.1016/j.swevo.2025.101914>, 2025.
- [26] Hassen Bouzgou, Christian A. Gueymard, "Minimum redundancy – Maximum relevance with extreme learning machines for global solar radiation forecasting: Toward an optimized dimensionality reduction for solar time series," *Solar Energy*, vol. Volume 158, no. ISSN 0038-092X, <https://doi.org/10.1016/j.solener.2017.10.035>, pp. Pages 595-609, 2017.
- [27] Jiefeng Zhou, Zhen Li, Kang Hao Cheong, Yong Deng, "Limit of the maximum random permutation set entropy," *Physica A: Statistical Mechanics and its Applications*, vol. Volume 664, no. ISSN 0378-4371, <https://doi.org/10.1016/j.physa.2025.130425>, 2025.
- [28] Ting Ke, Mingzhu Meng, Mengyan Wu, Jianfeng Qin, "A robust maximal margin hypersphere support vector machine with pinball loss," *Engineering Applications of Artificial Intelligence*, vol. Volume 155, no. ISSN 0952-1976, <https://doi.org/10.1016/j.engappai.2025.111020>, 2025.
- [29] John Alasdair Warwicker, Steffen Rebennack, "Support vector machines within a bivariate mixed-integer linear programming framework," *Expert Systems with Applications*, vol. Volume 245, no. ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2023.122998>, 2024.
- [30] N. C. F. A. Gopi Battineni, "Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM)," *Informatics in Medicine Unlocked*, vol. 16, p. 100200, 2019.
- [31] Manuel Garcia-Villalba, Tim Colonius, Olivier Desjardins, Dirk Lucas, Ali Mani, Daniele Marchisio, Omar K. Matar, Francesco Picano, Stéphane Zaleski, "Numerical methods for multiphase flows," *International Journal of Multiphase Flow*, no. ISSN 0301-9322, <https://doi.org/10.1016/j.ijmultiphaseflow.2025.105285>, 2025.
- [32] Ju Liu, Ling-Wei Huang, Yuan-Hai Shao, Wei-Jie Chen, Chun-Na Li, "A nonlinear kernel SVM classifier via L0/1 soft-margin loss with classification performance," *Journal of Computational and Applied Mathematics*, vol. Volume 437, no. ISSN 0377-0427, <https://doi.org/10.1016/j.cam.2023.115471>, 2024.
- [33] Deepika Agarwal, Pitam Singh, M.A. El Sayed, "The Karush–Kuhn–Tucker (KKT) optimality conditions for fuzzy-valued fractional optimization problems," *Mathematics and Computers in Simulation*, vol. Volume 205, no. ISSN 0378-4754, <https://doi.org/10.1016/j.matcom.2022.10.024>, pp. Pages 861-877, 2023.
- [34] Masood Ahmad, Muhammad Ahsan, Zaheer Uddin, "Explicit solution of high-dimensional parabolic PDEs: Application of Kronecker product and vectorization operator in the Haar wavelet method," *Computers & Mathematics with Applications*, vol. Volume 186, no. ISSN 0898-1221, <https://doi.org/10.1016/j.camwa.2025.03.001>, pp. Pages 1-15, 2025.
- [35] Kamilov M.M., Tojiboev B.M., Ravshanov A.A, "IMAGE SEGMENTATION USING FOURIER TRANSFORMATIONS TO IDENTIFY HOMOGENEOUS EARTH SURFACE OBJECTS," *PROBLEMS OF COMPUTATIONAL AND APPLIED MATHEMATICS*, vol. No. 2/1(65), no. ISSN 2181-8460, pp. 31-37, 2025.

- [36] Pilsoo Lee, "Inferring particle distributions in two-dimensional space with numerical features based on generative adversarial networks (GANs)," Nuclear Engineering and Technology, vol. Volume 57, no. Issue 10, pp. ISSN 1738-5733, <https://doi.org/10.1016/j.net.2025.103681>, 2025.
- [37] Vasiliy A. Es'kin, Alexey O. Malkhanov, Mikhail E. Smorkalov, "Are two hidden layers still enough for the physics-informed neural networks?," Journal of Computational Physics, vol. Volume 537, no. ISSN 0021-9991, <https://doi.org/10.1016/j.jcp.2025.114085>, 2025.