SFE2D ver. 5.1

for windows 7/8/10/11

Full User Manual

SFES2D: A 2D spatial & spectral feature extraction & selection tool for geo-imaging

SFES2D harnesses the power of advanced, unsupervised source separation techniques, including principal component analysis (for variance maximization), independent component analysis (for kurtosis and negentropy maximization), continuous wavelet transforms, RGB color processing, and k-means clustering segmentation. These cutting-edge methods enable feature extraction, dimensionality reduction of hyperdimensional geoscientific datasets, and lineament extraction with Bayesian optimization.

This software was developed and released by:

Bahman Abbassi

Postdoctoral Fellow in Mining Engineering

Institut de Recherche en Mines et en Environnement (IRME)

Université du Québec en Abitibi-Témiscamingue (UQAT)

Adresse: IC113, 445, boul. de l'Université

Rouyn-Noranda (Québec) J9X 5E4

Telephone: +1 (819) 762 0961 ext. 4311

Cell Phone: +1 (819) 290 1800

July 2024

Contents

1. Introduction

SFES2D is a standalone 2D spectral feature extraction program based on the integration of principal component analysis (PCA), independent component analysis (ICA), and continuous wavelet transform (CWT). The program offers spatial and spectral source separation for multiple sources of geoscientific data sets. SFES2D also provides a graphic user interface that can demonstrate compiled features in RGB images and segmented images through a fast k-mean clustering algorithm to reduce the number of colors and compile pseudo-geologic maps from extracted features. Users can also pick a color range by manually selecting the color intensities with mouse clicks. In the new version of the software, a module for extracting geological lineaments is included with Bayesian hyperparameter optimization. The program's application is straightforward and for immediate use as long as users already installed the MATLAB runtime library (free to download and install) on their computer.

2. System requirements

This program is designed to run on any Windows-based personal computer with at least 8 GB of random-access memory (RAM). Increasing the RAM size allows larger images to be processed at once. Since large matrices are operating in this program, the read/write speed of the storage is also essential. A solid-state drive (SSD) with a non-volatile memory express interface (NVMe) is recommended.

The SFES2D is provided in two formats: MATLAB P-codes and a standalone executable program.

P-codes require MATLAB 2024a version 24.1 (and later) to run. To do so, locate the P-codes in the current folder of MATLAB and then type SFES2D in the MATLAB Command Window.

- To use the executable program (SFES2D.exe), the user does not need any previously installed MATLAB software on the PC. The only prerequisite is the latest MATLAB 2024a Runtime library. Users can install these required runtime codes by double-clicking on the offline program installer “Installer_SFES2D (Offline).exe” or the online version of it “Installer_SFES2D (Online).exe” is a lightweight version in which the program and the runtime codes are going to be installed automatically through downloading from MathWorks server.

Either way, after the double click, the following windows pop up:

Figure 1

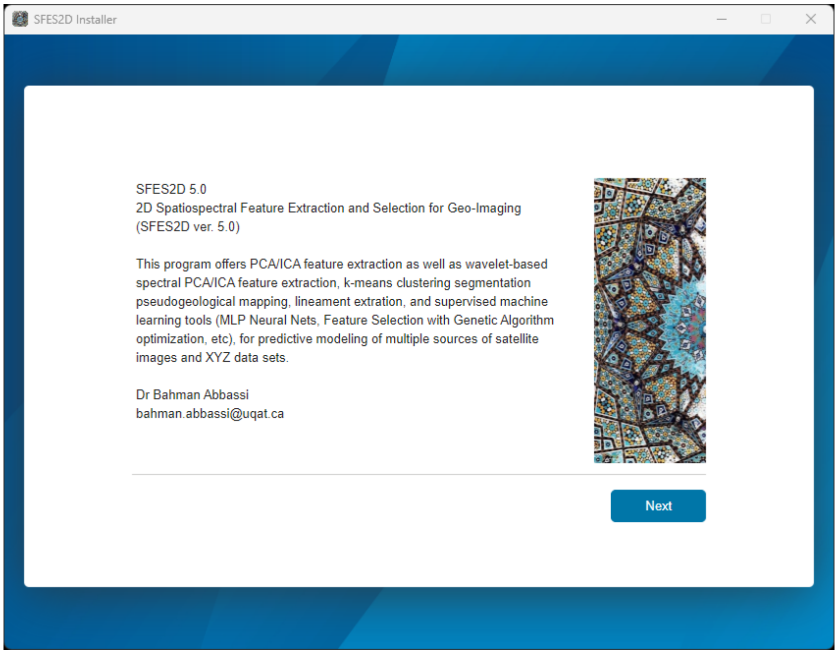


Figure 2

Press next to define the installation location for extracted files:

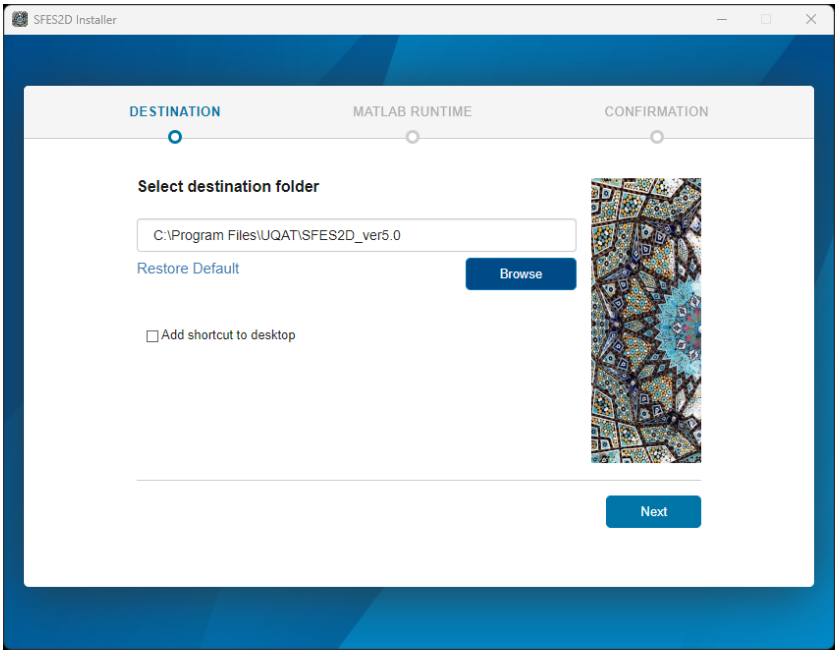


Figure 3

And then the location of MATLAB Runtime files:

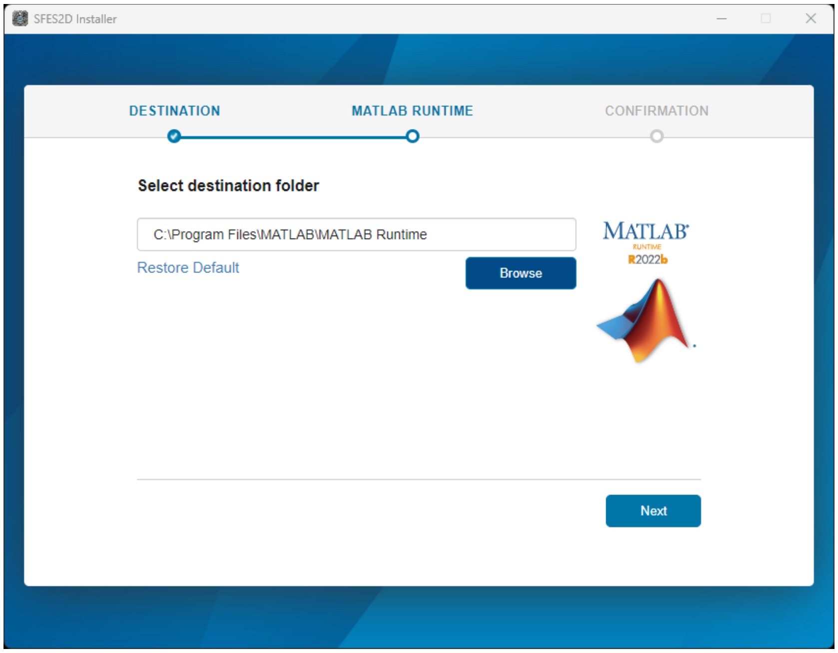


Figure 4

After accepting the terms of the license agreements, pressing "Install" should finalize the installation process:

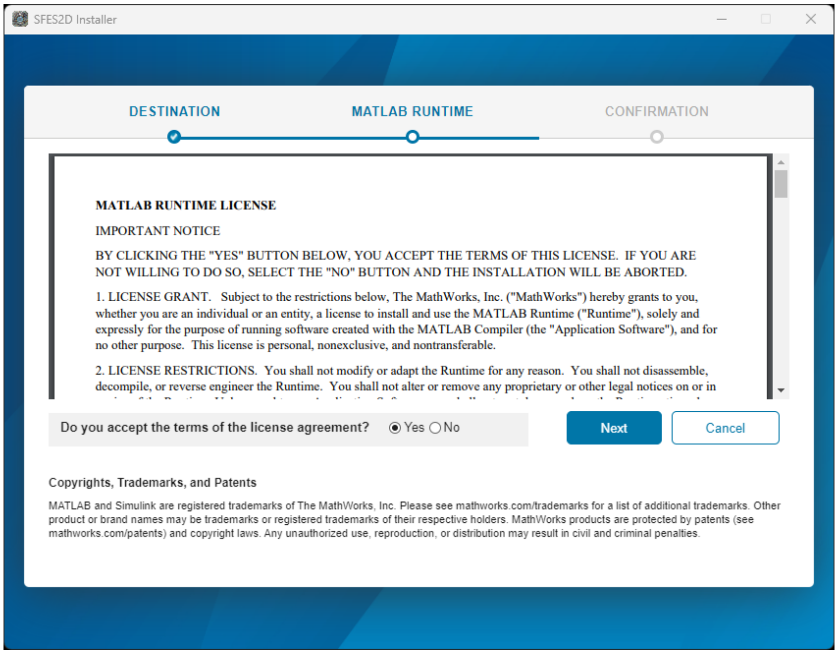


Figure 5

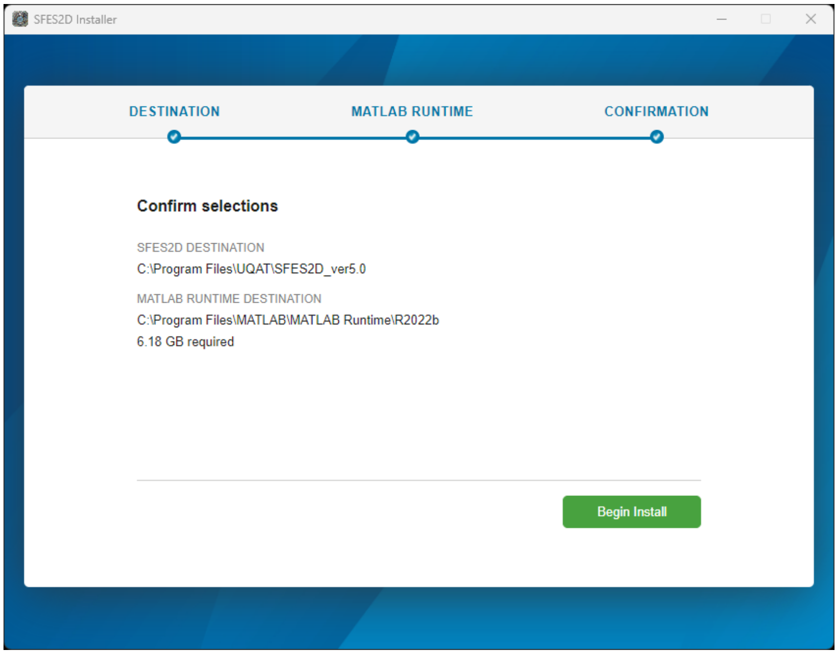


Figure 6

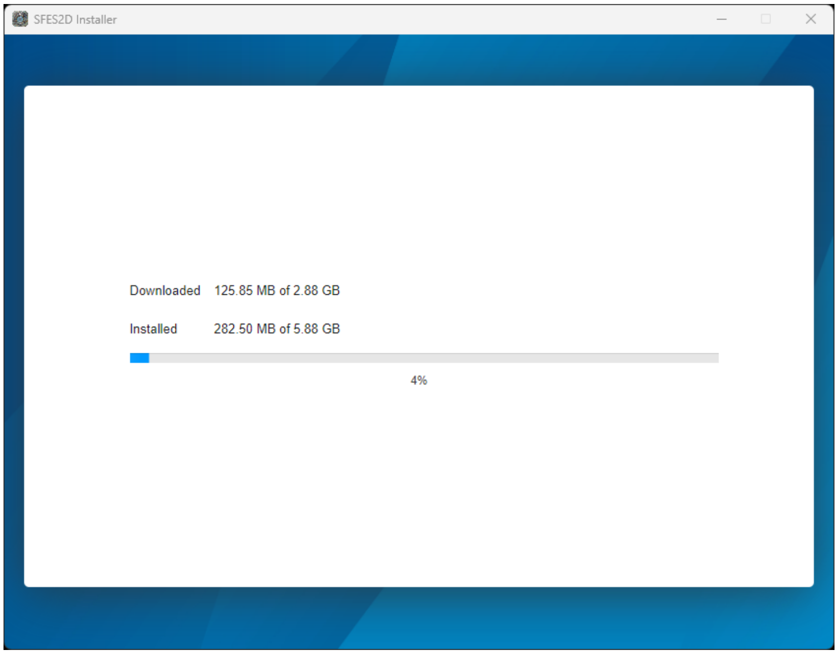


Figure 7

Users can also directly download the runtime from the address presented below:

https://www.mathworks.com/products/compiler/matlab-runtime.html (ver. 24.1).

After unzipping the file and installing the runtime, the program should work by double-clicking on the SFES2D.exe. It is recommended that the SFES2D.exe be copied/pasted to a project folder where data sets are located with read/write permission.

Running the program for the first time requires activation. To activate the program, double-click the Activate.exe file to create a Key file. A dialogue box (Password Required) appears that asks to insert the password provided for the user (you can get the password from the author by emailing bahman.abbassi@uqat.ca).

Figure 8

A key file will be generated when you type it and press Enter. The user must keep and save that file in the same project folder.

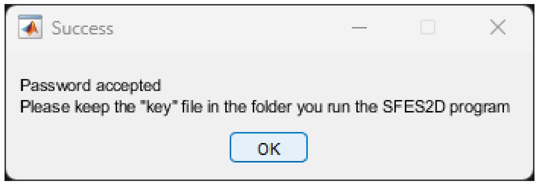


Figure 9

When running the SFES2D, the program interface appears as below:

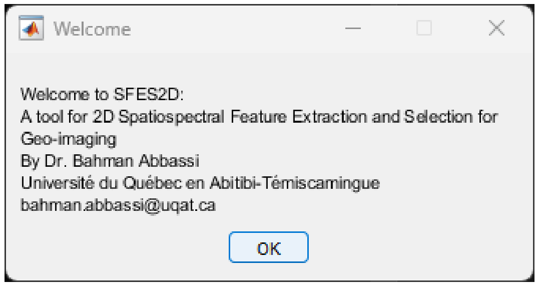


Figure 10

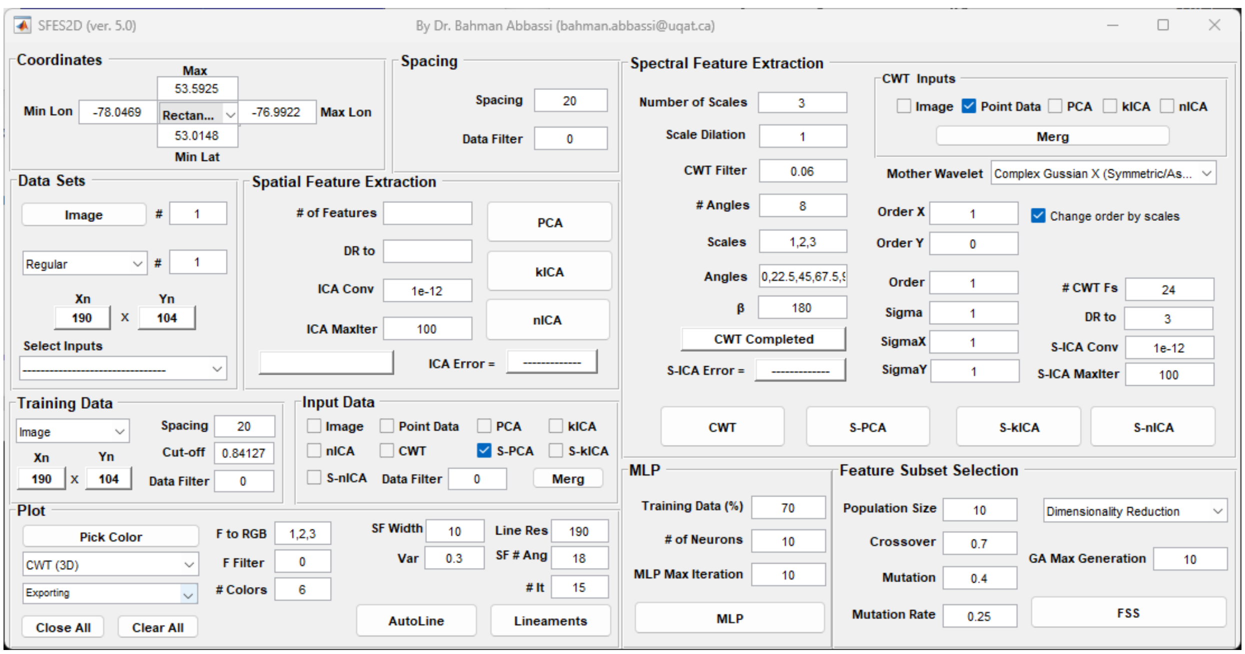


Figure 11

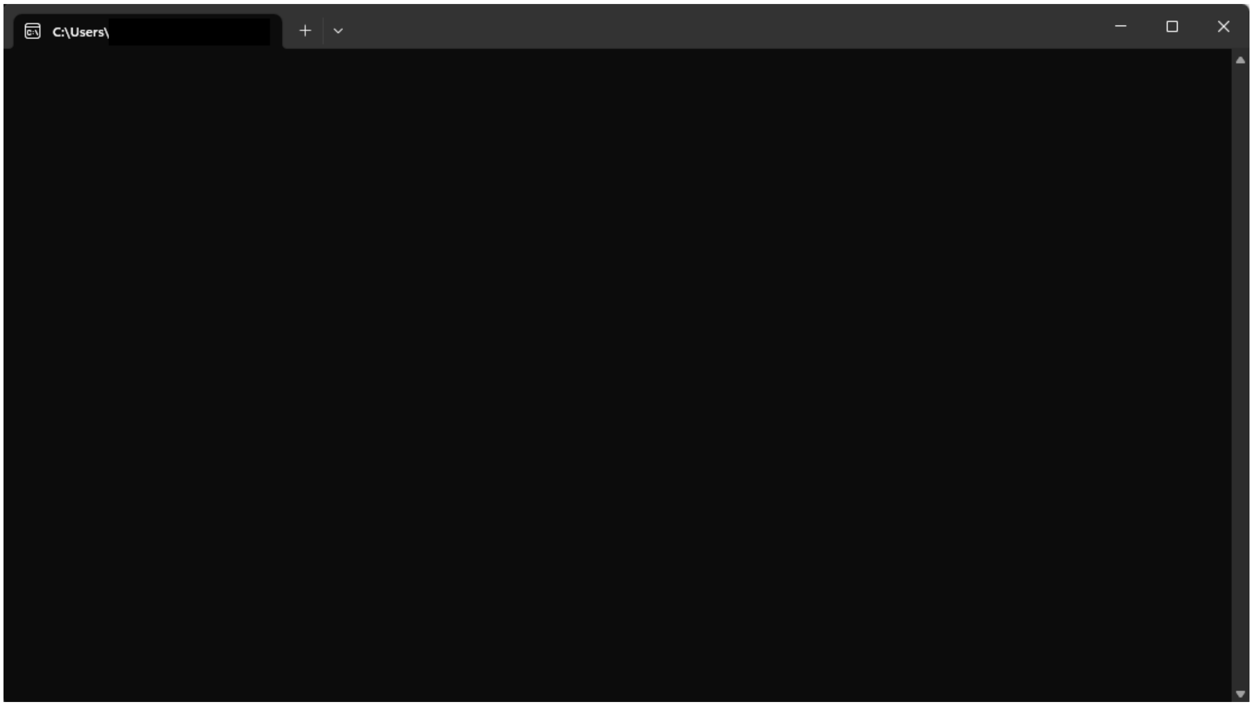


Figure 12

The user can control the program parameters at the top, and at the bottom, the progress of calculations and possible errors can be tracked with the Windows command shell console for execution (black screen).

3. Copyright

4. Theory

Multivariate feature extraction is a challenge and crucial for the automated interpretation of multiple geoscientific images (geo-images). In exploration geophysics, managing multiple data sources presents a significant pattern recognition challenge, necessitating the extraction of relevant features from numerous 2D/3D geo-images. However, the complexity of this task is amplified by the presence of complex "shadow effects" that often add heterogeneous overprints to these geo-images, making identifying latent features a daunting task. These shadow effects can stem from vegetation, water, snow, urbanization, clouds, pollution, rock compositions, and geochemical alteration. Such overprints are typically present in most geoscientific data sets; some are treated as noise, while others are considered valuable targets depending on the exploration's purpose.

This challenge can be described as a "hyperdimensional overlapping" problem. Various techniques exist for feature extraction in high-dimensional space, raising the question of how to detect latent features from large sets of hyperdimensional geo-images. PCA and ICA algorithms are useful for spatial source separation and dimensionality reduction. However, some images contain unique features that are not present in other geo-images. In these cases, spectral decomposition provides a distinctive feature extraction method in the frequency domain (Abbassi 2018).

This program introduces a 2D feature extraction scheme that enhances how we extract spatial/spectral features from multiple images. The SFES2D scheme, a product of extensive research and development, integrates 2D CWT with variance, kurtosis, and negentropy maximization algorithms. It also offers advanced tools for curvilinear pattern recognition (geological lineaments extraction) through wavelet and PCA dimensionality reduction, coupled with Bayesian hyperparameter optimization. Additionally, it includes state-of-the-art algorithms for supervised machine learning using a rapid multilayer perceptron algorithm and feature subset selection via bi-objective genetic algorithm optimization.

4.1. Spatial Feature Extraction

This section of the user manual details the theoretical framework and methodology employed for spatial feature extraction in geoscientific data analysis. The focus is on using blind source separation (BSS) methods to recover latent features from mixed geo-images, with an emphasis on Principal Component Analysis (PCA) and Independent Component Analysis (ICA).

4.1.1. Mixing Model

Geo-images are composed of various latent features combined in different proportions. Mathematically, this can be represented as (Abbassi et al., 2022):

$$g_i = \sum_{j=1}^{n} a_{ji}\, f_j \qquad (1)$$

where g_i denotes the observed geo-image, f_j represents the latent features, and a_{ij} are the mixing weights. The objective is to estimate the separation matrix W that can unmix the geo-images to recover these latent features.

Estimating a relevant separation matrix is possible by making assumptions about statistical measures such as correlation. In some instances, it is common for two geo-images to be statistically correlated. For example, suppose a linear relationship exists between two gamma-ray concentration images (e.g., K vs. eTh). In that case, the information in the first image is the same as in the second. Therefore, one can transform the bivariate data to a univariate form without losing valuable information. This transformation is called dimensionality reduction, the basis of Principal Component Analysis (PCA). PCA algorithms utilize the maximization of second-order statistical measures (variance) for image separation and produce linearly uncorrelated images. However, when there is a nonlinear form of correlation (dependency) between images, PCA will not work. In this case, ICA is applicable for separating geo-images into nonlinearly uncorrelated images by maximizing non-gaussianity.

4.1.2. Principal Component Analysis (PCA)

PCA is used to reduce the dimensionality of the data by transforming it into a set of linearly uncorrelated variables called principal components. This process involves several key steps:

1. Centering the Data:
    - Subtract the mean of each geo-image from the data to produce centered geo-images.
    - Mathematically, if g is the original geo-image and g_m is the mean, the centered image is $g - g_m$.

2. Whitening the Data:
    - Whitening transforms the centered data into unit variance and an identity covariance matrix.

- This is achieved through eigenvalue decomposition of the covariance matrix of the centered geo-images.
- The covariance matrix C is decomposed as $C = EDE^T$, where E is the matrix of eigenvectors and D is the diagonal matrix of eigenvalues.

3. Calculating the Transformation Matrix D:
   - The transformation matrix D converts centered images into whitened images.
   - The transformation is given by:

$$u = D(g - g_m) \tag{2}$$

   - In this context, D can be expressed as:

$$D = ED^{-1/2}E^T \tag{3}$$

   where $D^{-1/2}$ is the diagonal matrix with the inverse square root of the eigenvalues.

4. Deriving Principal Components:
   - The principal components are the transformed variables that maximize the variance of the data.
   - These components are linearly uncorrelated and are ordered by the amount of variance they capture from the data.

4.1.3. Independent Component Analysis (ICA)

ICA is used to separate geo-images into independent components by maximizing non-gaussianity. It involves two steps:

1. Whitening: As in PCA, to ensure the data has unit variance.

2. Rotation: Find a rotation matrix R that maximizes the non-gaussianity of the whitened images.

The ICA transformation can be represented as:

$$f = Ru \tag{4}$$

where u are the whitened images.

In SFES2D, two methods are used to obtain the rotation matrix *R*: Kurtosis Maximization (kICA) and Negentropy Maximization (nICA).

Kurtosis maximization uses the central limit theorem to measure the Gaussianity of the probability density functions. Kurtosis, the fourth-order cumulant of the whitened images, is expressed as the normalized fourth moment:

$$kurt(\mathrm{y}) = E\{\mathrm{y}^4\} - 3 \quad (5)$$

where $E\{\cdot\}$ denotes the expectation over the unknown density of input geo-images. Kurtosis provides a measure of how Gaussian (*Kurt* = 0), super-Gaussian (*Kurt* > 0), or sub-Gaussian (*Kurt* < 0) the probability density functions of the geo-images are.

The algorithm for kurtosis maximization in the SFES2D program incorporates deflationary orthogonalization to estimate independent components individually. The steps are:

1. Center and whiten the geo-image data to obtain *u*.
2. Initialize the Number of Independent Components *N*. Set a counter *i*.
3. Set an Initial Vector Randomly *w*. Ensure the vector is of unit norm.
4. Iteratively Update the Vector:

    — Perform rotation by deflationary orthogonalization

    $$w \leftarrow w - \sum_{j=1}^{i-1} (w^T w_j) w_j \quad (6)$$

    — Normalize the vector

    $$w \leftarrow w / \|w\| \quad (7)$$

5. Convergence Check: Repeat the rotation and normalization steps if the vector *w* has not converged.
6. Estimate the Separation Matrix:

    w is constructed from the converged vectors.

7. Obtain the latent features (Independent Components) by:

$$f \leftarrow wg \tag{8}$$

Negentropy is a measure of non-Gaussianity based on entropy. The entropy H of a geo-image g with a probability density p(g) is defined as:

$$H(\mathrm{g}) = -\int p(\mathrm{g}) \log p(\mathrm{g}) d\mathrm{g} \tag{9}$$

The negentropy J(g) is the difference between the entropy of a Gaussian random vector and the entropy of the geo-image:

$$neg(\mathrm{g}) = H\left(\mathrm{g}_{gauss}\right) - H\left(\mathrm{g}\right) \tag{10}$$

where g_{gauss} is a Gaussian random vector with the same covariance matrix as g.

Negentropy is always non-negative and zero when the signal has a Gaussian distribution. It can be approximated using higher-order cumulants. Since entropy estimation is not directly performed on geo-images, centered/whitened images (principal components) are used for entropy calculations. Therefore, g is replaced by y. For m higher order moments, we have:

$$neg(g) \approx \sum_{i=1}^{m} k_i \left[E\left\{G\left(\mathrm{y}_i\right)\right\} - E\left\{G\left(v\right)\right\} \right] \tag{11}$$

where v is a standardized Gaussian variable, k_i are constants, and GGG is a non-quadratic function that grows slowly (e.g., $G\left(\mathrm{y}\right) = -e^{\left(-\mathrm{y}^2/2\right)}$).

The algorithm for negentropy maximization in the SFES2D program is as follows:

1. Center and whiten the geo-image data to obtain u.

2. Initialize the Number of Independent Components N. Set a counter i.

3. Set an Initial Vector Randomly w. Ensure the vector is of unit norm.

4. Iteratively Update the Vector:

   — Perform rotation by deflationary orthogonalization:

$$w \leftarrow w - \sum_{j=1}^{i-1} (w^T w_j) w_j \qquad (12)$$

— Normalize the vector:

$$w \leftarrow w / \|w\| \qquad (13)$$

5. Update the vector using the fixed-point iteration for negentropy maximization:

$$\hat{W}_I \leftarrow E\left\{yg\left[\hat{W}_I^T y\right]\right\} - E\left\{g'\left[\hat{W}_I^T y\right]\right\}\hat{W} \qquad (14)$$

where G is the chosen non-quadratic function, and G′ is its derivative.

6. Convergence Check: Repeat the rotation and normalization steps if the vector *w* has not converged.

7. Estimate the Separation Matrix: *w* is constructed from the converged vectors.

8. Obtain the latent features (Independent Components) by:

$$f \leftarrow wg \qquad (15)$$

4.2. Spectral Feature Extraction

Spectral feature extraction in the frequency domain is achieved through the spectral decomposition of geo-images. Unlike the Fourier transform, which struggles with abrupt changes, wavelet transforms are localized in space and have finite durations, making them ideal for capturing sharp image changes (Antoine, et al., 2004).

The Continuous Wavelet Transform (CWT) decomposes a signal/image into scaled and translated versions of a chosen wavelet (mother wavelet). First applied in geophysical data analysis in 1997, CWT is pivotal for locating potential field sources by identifying maxima lines derived from CWT coefficients.

A 2D wavelet is a natural or complex-valued oscillatory function adhering to the admissibility condition within the real plane. The Fourier transform of any pertinent wavelet function must be square-integrable and possess a zero-mean value. This ensures finite energy in the spatial and frequency domains, favoring spatial and frequency localization, multi-resolution analysis, and efficient delineation of low and high-frequency signals.

Given a 2D image *I*, the CWT coefficients can be expressed as:

$$C_I\left(a,\vec{b},\theta\right)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}I(x,y)\psi^*\left(\frac{x-b_x}{a}-\frac{y-b_y}{a},\theta\right)dxdy \qquad (16)$$

a is the scale factor, b is the translation vector, θ is the rotational angle, and ψ^* complex conjugate of the mother wavelet. The wavelet coefficients measure the similarity between the image and the scaled, translated, and rotated mother wavelet.

The SFES2D program integrates various wavelet algorithms, including:

1. Difference-of-Gaussian (Symmetric)
2. Complex Gaussian Z (Asymmetric/Symmetric)
3. Complex Gaussian X (Asymmetric/Symmetric)
4. Derivatives of Gaussian (Asymmetric)
5. Mexican Hat (Symmetric)
6. Morlet (Asymmetric)
7. Isotropic Morlet (Symmetric)
8. Cauchy (Asymmetric)
9. ES-Cauchy (Asymmetric)
10. Paul (Symmetric)
11. Wheel (Symmetric)
12. Pethat (Symmetric)
13. End-Stop (Asymmetric)
14. Gabor Mexican Hat (Asymmetric)
15. Sinc (Symmetric)
16. Fan (Symmetric)

Spectral decomposition produces high-dimensional images with patterns linked to specific scales and directions, making some decomposed spectra redundant. Dimensionality reduction is necessary to extract the most relevant features from high-dimensional data, aiding in automated pseudogeological mapping, lineament analysis, and machine learning estimations.

The feature extraction scheme combines 2D CWT with variance, kurtosis, and negentropy maximization algorithms (PCA, kICA, and nICA) for spectral feature extraction. The algorithm consists of three main stages (Figure 13):

1. Preprocessing: 2D interpolation and filtering of raw datasets prepare the input images for feature extraction.
2. Spatial Feature Extraction: Employs PCA/ICA for spatial source separation and dimensionality reduction.
3. Spectral Feature Extraction: Consists of two substages:

   a. Continuous wavelet transform (CWT)

   b. Spectral PCA/ICA (SPCA/SkICA/SnICA)

Effective spectral feature extraction depends on computer hardware specifications, particularly for large numbers of spectral features. Users can create more features by adjusting CWT parameters. For large numbers of CWT features, PCA/ICA methods help decorrelate and separate spectral features. Dimensionality reduction techniques can summarize features, retaining the most geologically pertinent information.

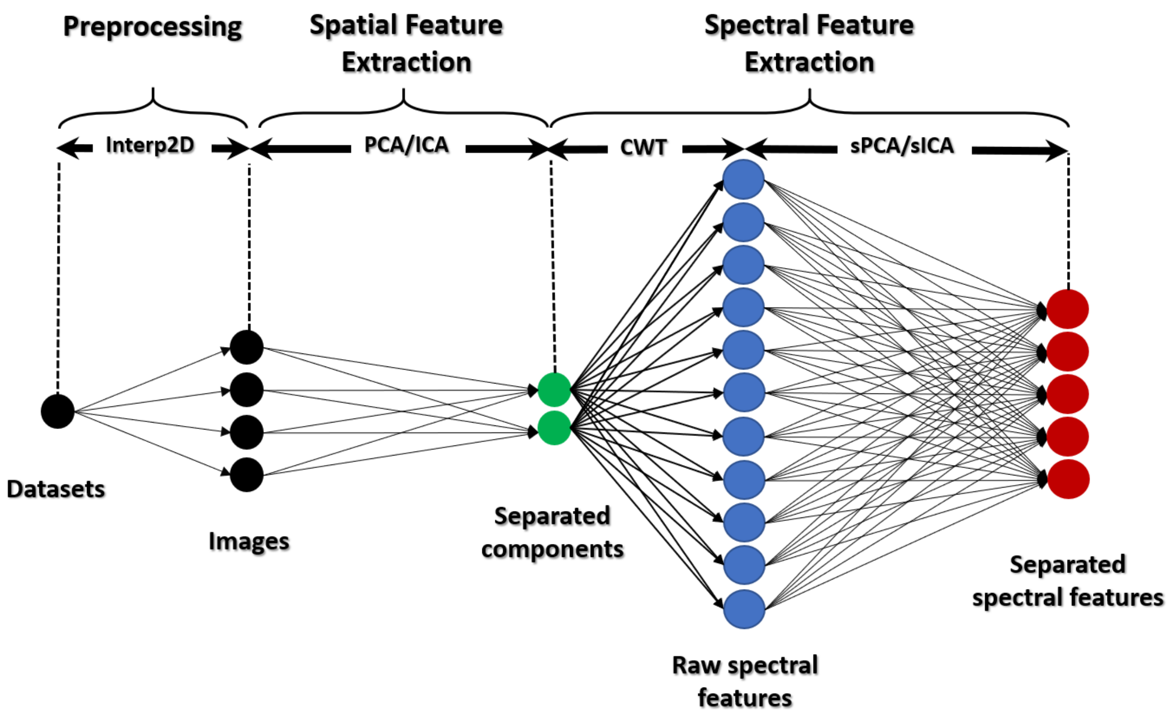


Figure 13. Schematic view of the SFES2D procedure.

4.3. Feature representation

The SFES2D proposes feature representation tools focusing on RGB image compilation, color pick algorithm, and segmentation analysis (Abbassi and Cheng, 2021).

4.3.1. RGB Image Compilation and Analysis

Users can select the RGB image compilation method to assemble a colored image from three manually chosen extracted features using PCA and ICA methods. The RGB combination helps highlight hidden characteristics in the data. Users can graphically set the label number of desired features for image compilation. For example, in nICA feature extraction, labels 1, 2, and 3 mean the features 1, 2, and 3 are used for image compilation. The polarity of features can be adjusted since PCA/ICA methods can distort polarities randomly (e.g., setting -2, 1, and 3 for RGB compilation).

4.3.2. Color Pick Algorithm

The SFES2D program includes a color pick algorithm for selecting regions of interest (ROI) based on color intensities. The algorithm assigns low and high thresholds for each RGB color band based on user clicks on regions of interest. More clicks on specified zones result in more accurate thresholding. The program keeps a range of 5 to 95 percentiles to reduce the effect of outliers, resulting in smoother color-picking tasks.

4.3.3. Image Segmentation Algorithm

The SFES2D program employs a k-means clustering algorithm for segmenting color images, essential for reducing the color space to a manageable number of colors. This segmentation facilitates the creation of pseudo-geological maps, aiding geologists in identifying hidden geological structures within geo-images. It also enhances memory efficiency and boosts image analysis by concentrating on relevant information.

1. Initial Centroid Generation

   Randomly generate k initial centroids:

   $\{c_1,c_2,\ldots,c_k\}\leftarrow$random initialization.

   These centroids represent the initial guesses for the centers of the k clusters.

2. Assign data points to the nearest centroid:

   For each data point p, calculate the distance to each centroid c_i. Assign each data point ppp to the cluster with the nearest centroid. This step results in k clusters $\{S_1,S_2,\ldots,S_k\}$:

$$S_i \leftarrow \left\{ p \colon \left\| p - c_i \right\|^2 \le \left\| p - c_j \right\|^2, \quad \forall\, j \ne i \right\} \tag{17}$$

   where ‖·‖ denotes the Euclidean distance.

3. Update Centroids

   Recalculate the centroids by taking the mean of all data points assigned to each cluster. Update the position of each centroid to this mean value. The new centroids minimize the within-cluster sum of squares, improving cluster cohesion:

$$c_i \leftarrow \frac{1}{|S_i|}\sum_{p\in S_i} p \quad (18)$$

4. Calculate Euclidean Distance in RGB space

   Calculate the 3D Euclidean distance between each data point and the centroids in RGB images.

$$\|p-c_i\| = \sqrt{(R_p - R_{c_i})^2 + (G_p - G_{c_i})^2 + (B_p - B_{c_i})^2} \quad (19)$$

   This distance metric helps accurately cluster color data points based on their RGB values.

Benefits of color segmentation by SFES2D include:

- Memory Efficiency: The algorithm reduces memory usage by focusing on segmented regions.
- Speed: Segmentation accelerates image analysis by concentrating on the most relevant features.
- Pseudo-Geological Maps: The segmented images facilitate the creation of pseudo-geological maps, which highlight geological contacts and structures, aiding interpretation.

4.4. Lineament extraction with Wavelet-ICA and Bayesian Hyperparameter Optimization

SPES2D ver. 5.1 introduces a comprehensive algorithm for extracting geological curvilinear lineaments from geophysical and geoscientific data sets. It combines several advanced methods, including Continuous Wavelet Transform (CWT, refer to section 4.2), Principal Component Analysis (PCA, refer to section 4.1), Independent Component Analysis (ICA, refer to section 4.1), hysteresis thresholding, and Bayesian Hyperparameter Optimization (BHO).

CWT decomposes an image into scaled and translated versions of a wavelet, providing detailed space-frequency information. This increases dimensionality, which PCA addresses by retaining significant variance while reducing dimensionality. This facilitates the detection of linear patterns through hysteresis thresholding.

4.4.1. Hysteresis Thresholding algorithm

The Hysteresis Thresholding algorithm detects linear patterns by computing slopes and aspects of spectral features, enhancing the image, and pixel labeling to map potential lineaments.

The Hysteresis Thresholding algorithm is as follows:

1. Preprocessing
    - Image Preparation: Normalize the color distribution and apply low-pass filtering to remove noise.
    - Wavelet Transform: Perform 2D CWT on the preprocessed images to decompose them into multiple scales and directions, increasing the dimensionality.
    - Dimensionality Reduction: Apply PCA to the wavelet-transformed images to reduce dimensionality while preserving significant variance.
2. Slope and Aspect Calculation
    - Compute the gradient of the principal component P at each point (x,y):

$$\nabla P = \left(\frac{\partial P}{\partial x}, \frac{\partial P}{\partial y} \right) \qquad (20)$$

- Calculate the slope S and aspect A from the gradient:

$$S = \|\nabla P\|,\ A = \arctan\left(\frac{\partial P / \partial y}{\partial P / \partial x} \right) \qquad (21)$$

- Enhanced images L are created by identifying areas with high slopes, slopes of slopes, and slopes of aspects (Panagiotakis and Kokinou, 2014 and 2015):

$$L = \left(S^2 . S' . A' \right)^{1/4} \qquad (22)$$

S' and A' are the "slope of the slope" and the "slope of the aspect" respectively.

3. Hysteresis Thresholding
    - Define Thresholds:
        - Low threshold T_l to identify weak lineament pixels.
        - High threshold T_h to identify strong lineament pixels.
    - Improve curvilinear patterns:
        - Convolves L with a zero mean step filter G with a width of w and an orientation angle φ:

$$I_g = L * G(w, \varphi) \tag{23}$$

- Then the algorithm calculates the maximum of the corresponding pixel values of the images:

$$I_m = \max_{a,\varphi} I_g \tag{24}$$

— Calculate k, the median value of the 9-pixel neighborhood of each pixel p in the enhanced image I_m.

— Classify pixels:

- Strong lineament pixel (C_1): If $I(p) > T_h$ and $I(p) > k$.
- Weak lineament pixel (C_2): If $T_l < I(p) \leq T_h$ and $I(p) > k$.
- Non-lineament pixel (C_3): If $I(p) \leq T_l$ or $I(p) \leq k$

— Region Growing:

Classify C_2 pixels as C_1 if connected to a C_1 pixel; otherwise, classify them as C_3.

— Final Lineament Map:

Combine the results to produce a binary image highlighting the potential faults.

4.4.2. Bayesian hyperparameter optimization

Bayesian Hyperparameter Optimization (BHO) is a crucial component in the lineament extraction process within the SFES2D program. It systematically optimizes hyperparameters to enhance the extraction of geological lineaments from geophysical data sets. BHO combines probabilistic modeling and iterative improvement to explore the hyperparameter space efficiently. The optimization considers hyperparameters related to wavelet smoothness, PCA dimensionality reduction, and hysteresis thresholding.

The primary goal is to maximize the F_β Score, which balances precision and recall in the extraction process. The steps of the BHO algorithms are as follows:

1. Define the Objective Function

   Formulate the objective function based on the Fβ Score

2. Initialize the Gaussian Process (GP) Model

   The Gaussian Process is a probabilistic model that estimates the objective function. Defined by a mean function $\mu(x)$ and a Matérn kernel function $k(x, x')$:

$$f_{obj}(x) \sim GP\left(\mu(x), k(x, x')\right) \tag{25}$$

3. Generate Initial Data Points

   Start with N randomly selected initial hyperparameter sets forming the initial dataset D_0.

4. Iterative Optimization Process

— Fit GP Model:

   Fit the Gaussian Process to the initial dataset D_{t-1} to model the objective function.

— Optimize Acquisition Function:

   Use the Expected Improvement (EI) acquisition function to balance exploration and exploitation. It guides the selection of the next hyperparameter set x_t:

$$EI(x) = \mathrm{E}\left[\max\left(f(x) - f(x^+), 0\right)\right] \tag{26}$$

   where $f(x^+)$ is the current best-known objective value.

— Evaluate Objective Function:

   Evaluate the objective function at x_t to obtain $f(x_t)$.

— Update dataset:

   Update the dataset to include the new observation $D_t = D_{t-1} \cup \{(x_t, f(x_t))\}$

5. Convergence Check
   - Check if the optimization has converged based on predefined criteria (e.g., minimal improvement over several iterations).
   - If not converged, return to the iterative optimization step.

6. Optimal Hyperparameter Selection:

   Once convergence is achieved, select the hyperparameter set with the highest F_β Score from the observations.

4.5. Machine Learning

The SFES2D program incorporates advanced machine learning techniques to enhance the extraction and analysis of geological lineaments from geophysical data sets. This approach is divided into two main sections: Spectral Representative Learning (SRL) and Spectral Feature Subset Selection (SFSS). Both sections focus on leveraging spectral decomposition and machine learning algorithms to identify and enhance significant features, improving the accuracy and efficiency of geological interpretation.

4.5.1. Spectral Representative Learning (SRL)

Spectral Representative Learning (SRL) utilizes spectral decomposition to break geophysical data into constituent spectral components. This process, primarily performed using CWT and PCA/ICA methods (for dimensionality reduction), allows for the detailed analysis of frequency-dependent features. SRL then involves machine learning algorithms, specifically a Multi-Layer Perceptron (MLP), to identify and represent the most relevant spectral features.

The algorithm steps of the SRL are as follows (Figure 14):

1. Preprocessing
   - Data Preparation: To enhance the images, prepare the data sets using conventional geophysical data processing, inversion, and filtering methods.
   - Interpolate the data to create 3D representations suitable for analysis.
2. Spatial Feature Extraction (section 4.1).
3. Spectral Feature Extraction (section 4.2).
4. Machine Learning with Multi-Layer Perceptron (MLP):

- Input: Takes the extracted spectral features as input.
- Hidden Layers: Multiple layers with weights W1 and W2 are adjusted during the learning process:

$$H^l = \sigma\left(W^l H^{l-1} + b^l\right) \tag{27}$$

  where H^l is the hidden layer at level l, W^l are the network weights, b_l are the biases, and σ is the activation function.
- Targets: Geological or other significant information, such as fault orientations, mineral deposits, structural features, metal grades, etc., must be predicted.
- Training: The network learns by minimizing the loss function F while adjusting the weights to optimize the prediction accuracy. The MLP is trained using a supervised learning approach to minimize the loss function F, the least mean square error:

$$F = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - t_i\right)^2 \quad (28)$$

Where y_i is the predicted output, t_i is the target output, and N is the number of training samples.

— Output Layer: Produces the final output based on the learned patterns. The output is produced based on the learned patterns from the hidden layers:

$$y = \sigma\left(HW1 + b_1\right)W2 + b_2 \quad (29)$$

Where y is the output vector, $W1$, and $W2$ are the weight matrices of the hidden layers, and b_1 and b_2 are the biases.

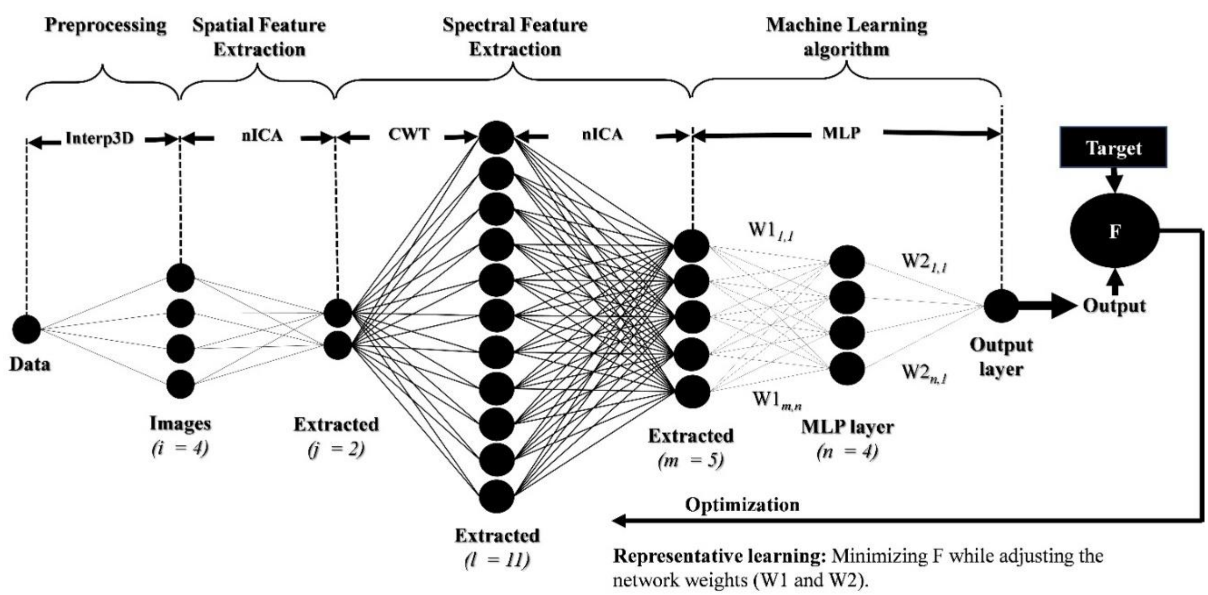


Figure 14. Schematic view of SRL for predictive modeling. The algorithm consists of four sub-modules: The preprocessing prepares the data sets by 3D inversion, interpolation, and filtering methods. The spatial feature extraction with nICA separates the image overlaps in 3D. Spectral feature extraction with CWT-nICA extracts the wavelet decomposed features. Furthermore, the machine learning algorithm with MLP adjusts network weights (W1 and W2) to learn patterns inside the extracted features based on the sample targets.

4.5.1. Spectral Feature Subset Selection (SFSS)

Spectral Feature Subset Selection (SFSS) is an advanced method within the SFES2D program designed to optimize the selection of spectral features for improved geological interpretation. SFSS addresses the challenge of high-dimensional data by identifying the most relevant features, thereby enhancing the performance of machine learning models used for predictive modeling in geophysics.

The main objective of SFSS is to reduce the dimensionality of the spectral feature set, improve computational efficiency, and enhance the accuracy of predictions made by machine learning algorithms. SFSS aims to streamline the learning process and improve generalization by selecting a subset of the most informative features.

The algorithm steps of the SFSS are as follows (Figure 15):

5. Preprocessing

— Data Preparation: To enhance the images, prepare the data sets using conventional geophysical data processing, inversion, and filtering methods.

— Interpolate the data to create 3D representations suitable for analysis.

6. Spatial Feature Extraction (section 4.1).

7. Spectral Feature Extraction (section 4.2).

8. Machine Learning with Spectral Feature Subset Selection (SFSS)
    - Input: Takes the extracted features as input.
    - Hidden Layers: Multiple layers with weights W1 and W2 are adjusted during the learning process.
    - Targets: Geological or other significant information, such as fault orientations, mineral deposits, structural features, metal grades, etc., must be predicted.
    - Bi-Objective Genetic Algorithm (GA): The bi-objective GA minimizes the ANN cost function E_{ANN} and the number of selected features n_f:

$$\min\left(f = [E_{ANN}, n_f] \right) \quad (30)$$

This approach ensures that the selected features are relevant and contribute to the model's efficiency:

- Population Initialization

  Randomly initialize a population of potential solutions (feature subsets).
- Fast Non-Dominated Sorting (*NDS*)

  Sort the population into different non-dominated fronts based on their objective values, such as feature subset size and model performance:

$$NDS : \{F_1, F_2, ..., F_k\} \tag{31}$$

  where F_i represents a front in the sorted population.
- Crowding Distance Estimation

  Calculate the average distance between solutions in each front to maintain diversity in the population:

$$d_i = \frac{1}{k-1} \sum_{j=1}^{k-1} \left(f_{j+1} - f_{j-1} \right) \tag{32}$$

  where the d_i is the crowing distance for the *i*-th solution.
- Binary Tournament Selection

  Select individuals for mating based on their non-domination level and crowding distance.
- Crossover and Mutation

  To explore the feature space, generate offspring through simulated binary crossover and polynomial mutation operators.
- Recombination and Selection

  Combine parents and offspring, then select the best solutions for the next generation. Repeat the process for the desired number of generations to find the best multi-objective solutions.

— Output: Produces the final output based on the learned patterns, including the predicted values and the selected spectral features.

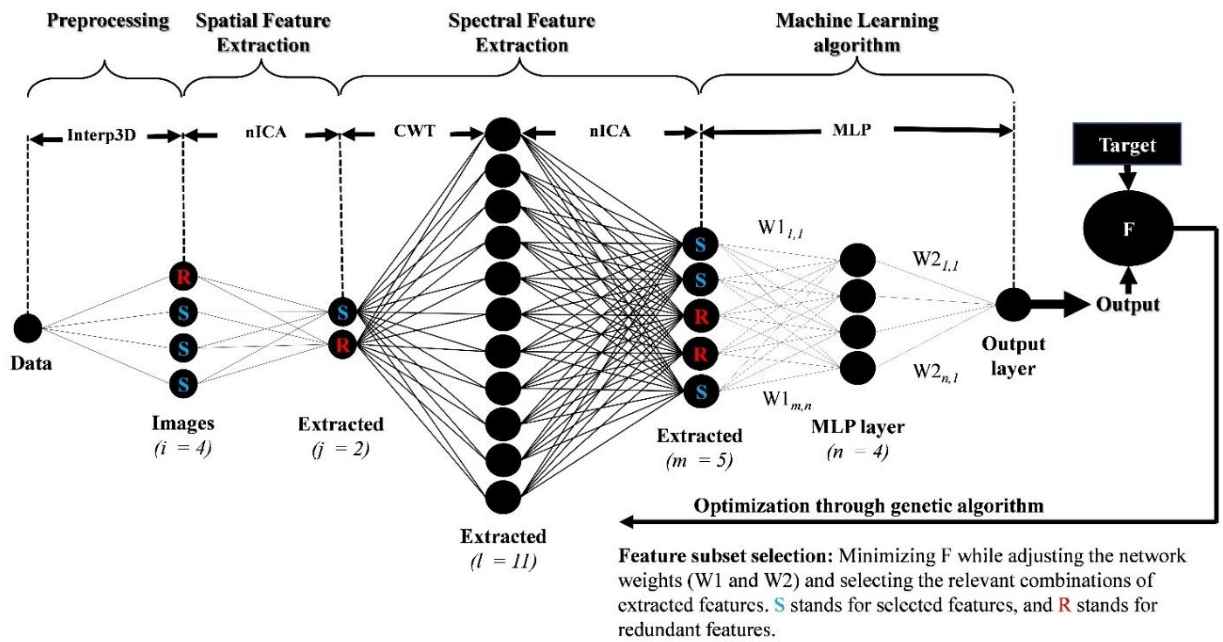


Figure 15. Schematic view of spectral feature subset selection for predictive modeling. The algorithm consists of four main sub-routines, such as the SRL algorithm, except that the MLP is integrated with NSGA-II to adjust the network weights and the selection of inputs simultaneously.

5. Input/output formats

After defining the minimums and maximums of the geographic coordinate system (latitude and longitude in decimal degrees) and the spacing (latitude and longitude in Arc-Second), the program supports two groups of inputs: RGB color images and point data sets. The color images can be read in BMP/PNG/JPG formats. The number of pixels is automatically adjusted depending on the project size (minimum/maximum of coordinates and spacing). The point data sets should be in the form of Comma-Separated Values (CSV) with three columns. The first column should be the longitude values (X), the second column should be the latitude values (Y), and the third column should be the measured/observed values (Z).

The output of the program can be in many forms:

- RGBs of separated components. Users have control over selecting the desired components to compile their models.
- Interactive selection of the region of interest (ROI) based on RGB colors.
- Color-segmented maps for zoning of different extracted features by a fast k-means clustering algorithm.
- All matrices are CSV and GRD files (Golden et al. format).

6. Program environment

The SFES2D program is designed for spatial and spectral feature extraction of unlimited geo-images. The program is made of six main windows, including:

- **Coordinates**: Preferably, the geographic coordinate system in SFES2D v.1.4.

Two methods are used for clipping the data sets:

A) Rectangular coordinates:

Coordinates are readable from a prepared text file with this format:

First line: Min Longitude (Min Lon)

Second line: Max Longitude (Max Lon)

Third line: Min Latitude (Min Lat)

Fourth line: Max Latitude (Max Lat)

For example, in the case above:

-77.8

-77.2

53.15

53.45

For satellite images, a separate text file is also needed to upload the original image coordinates for georeferencing purposes.

If you use microscopic images without georeferencing, you use a text file with this content if the image has a square size (x and y pixels are the same):

0

1

0

1

If not, consider the number of pixels in each direction. For example, if your image has an 1138 x 866 resolution, you need to use a ratio compatible with this resolution. The coordinate text file will look like this:

0

1.314

0

1

1.314 is the ratio of the input image's horizontal to vertical resolution (1138 /866). You need to reread the same text file when you upload the image.

B) Automatic polygonization:

Sometimes, geophysical data sets are geometrically irregular and will not cover any rectangular area. The program automatically polygonizes the 2D space for these cases based on irregular coordinates. You can also digitize the region of interest on any GIS software and use the XY coordinates for automatic polygonization in SFES2D. You must simply upload the CSV file related to that data set with at least two X and Y coordinate columns.

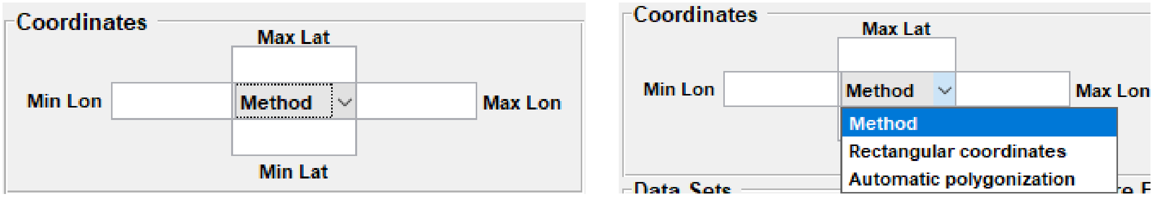


Figure 27

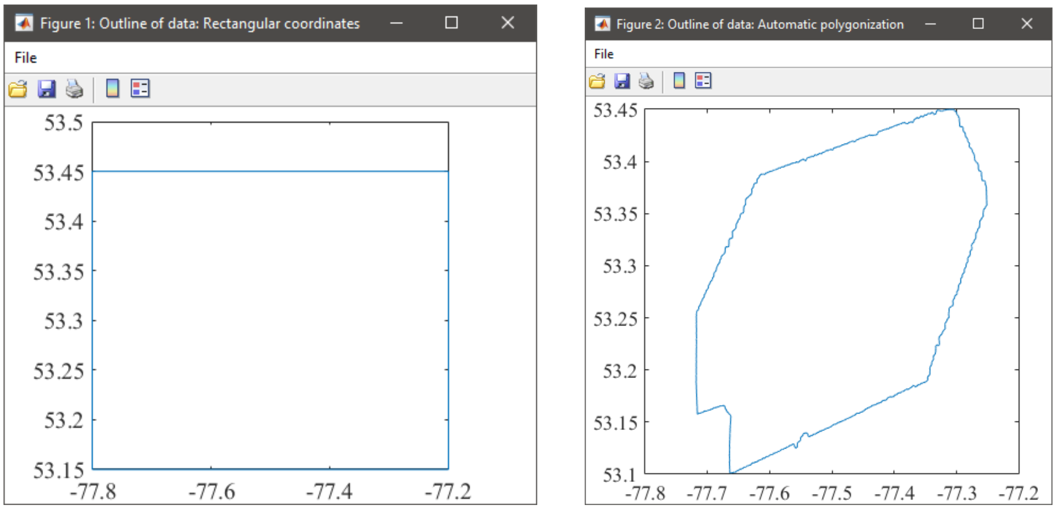


Figure 28

➢ **Spacing**: X and Y (Longitude and Latitude) spacings are in arcsecond. In Quebec, each Arc-Sec of latitude is about 33 meters, and each Arc-Sec of longitude is about 17 meters. The ratios are automatically adjusted for any location on Earth so that the overall scales of the images are affected for optimal presentation.

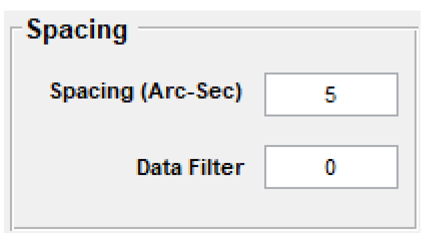


Figure 29

➢ **Data sets**: We can import RGB satellite images as well as point data sets in XYZ format (magnetic, gravity, elevation, gamma-ray concentrations, and so on). If needed, we can add support for hyperspectral imaging in the next versions.

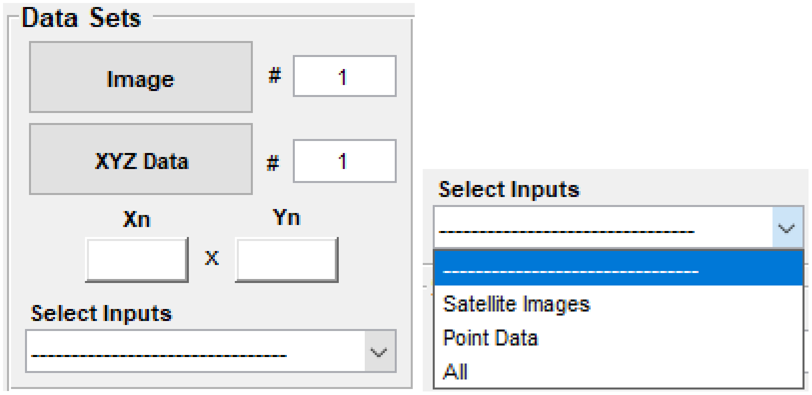


Figure 30

➢ **Spatial Feature Extraction**: For spatial feature extraction of multiple geo-images through PCA, kICA, and nICA. The number of features has to be equal to or less than the number of input geo-images (in the case of dimensionality reduction). Convergence and maximum iteration of the ICA methods are also available for users to change in the full version of the software. The demo version of the software is limited to 10 iterations and a threshold of 0.001 for convergence criteria.

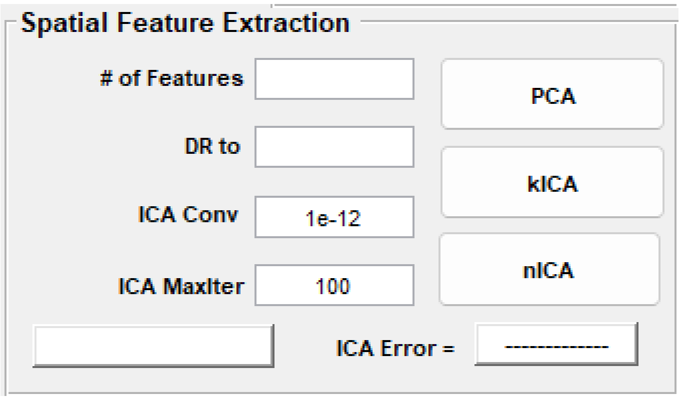


Figure 31

➢ **Spectral Feature Extraction**: This window consists of two major sections. Firstly, a 2D CWT decomposes desired inputs into raw spectral features. Users control the number of scales, scale dilation, and angles in which the mother wavelet will surf. Then, each mother wavelet structure automatically calculates the scales vector, corresponding frequencies, and angles vector.

Various isotropic and anisotropic mother wavelets are also presented here for user use based on the purpose of the studies. Secondly, S-PCA, S-kICA, and S-nICA are also available for spectral source separation. After CWT, the program automatically sets the number of CWT features. Users must decide whether dimensionality reduction is required (e.g., 100% for no dimensionality reduction). For spectral ICA methods (S-kICA and S-nICA), convergence and maximum iteration criteria are also needed. The demo version of the software is limited to 10 iterations and a threshold of 0.001 for convergence criteria.

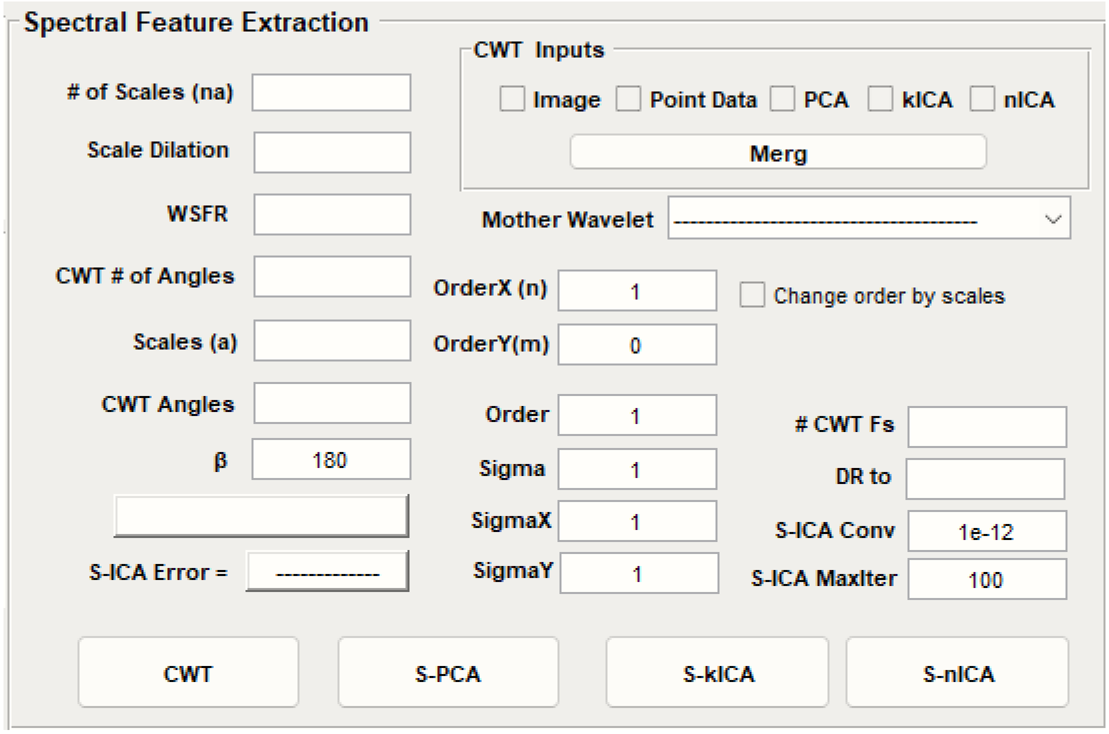


Figure 32

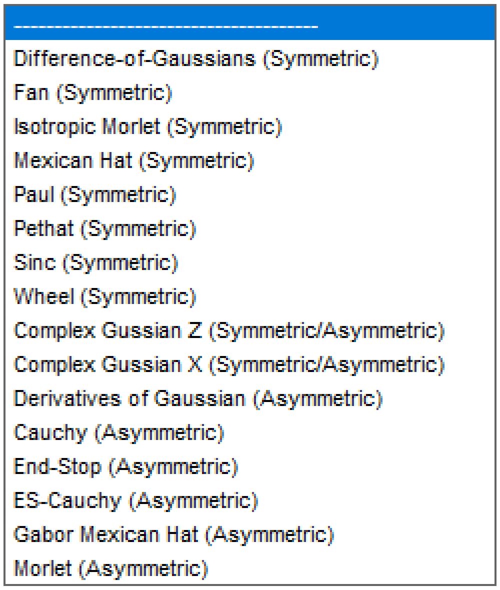


Figure 33

➢ **Plot**: This window is designed to illustrate and export the data sets and results. The program enables the user to pick the desired extracted features and integrate them in red/green/blue (RGB) images to visualize underlying features efficiently. Users can export results in XYZ and grid formats (.GRD) compatible with the Golden Software Surfer program.

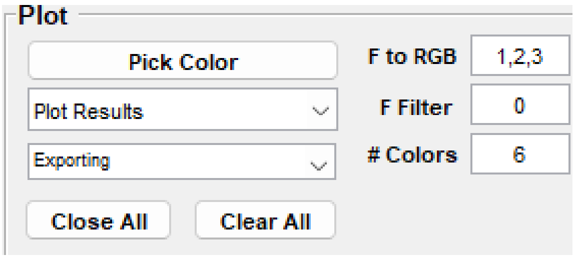


Figure 34

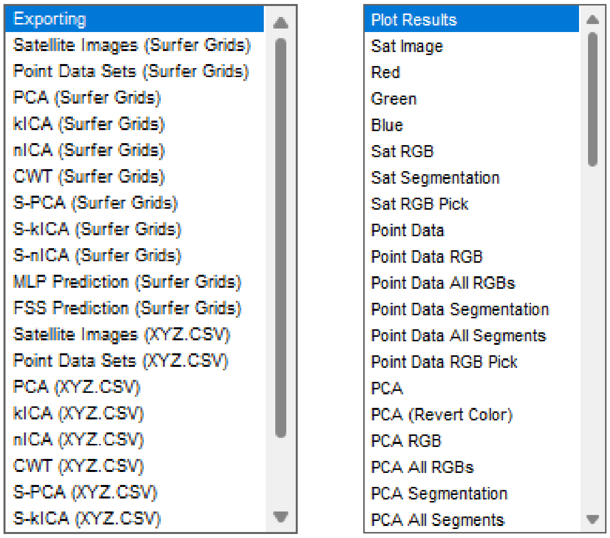


Figure 35

➢ **Lineaments**: This window is added in ver. 5.1 It is designed for lineament extraction and automated Bayesian optimization for lineament detection.

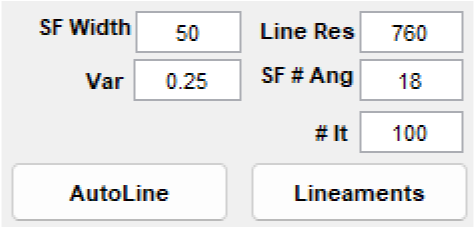


Figure 36

7. Spatial feature extraction in SFES2D

Spatial feature extraction refers to the application of blind source separation methods in the spatial domain to extract the previously mixed features from multiple geo-images. Dimensionality reduction is also possible. A standard way to perform a spatial feature extraction in SFES2D is as the following workflow:

- Read the coordinates.

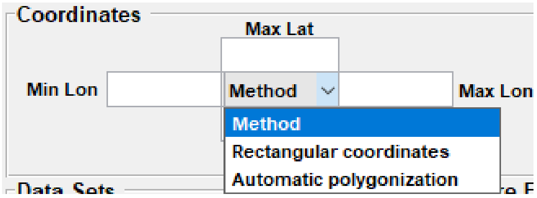


Figure 37

- Set Spacing and Data Filter values. If you do not want to smooth the data sets, just insert zero in the box any value greater than 0 will smooth the data sets proportional to the magnitude of the Data Filter.

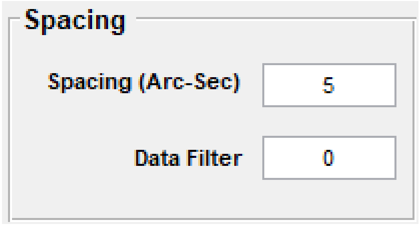


Figure 38

- Push *Image/XYZ Data* buttons in the *Data Sets* module and select the desired data sets.

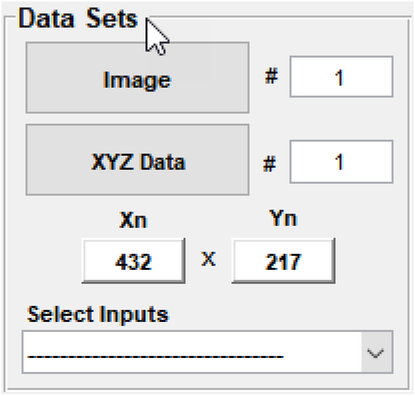


Figure 39

- To insert more images or XYZ data, change the value in the # box on the right to 2, 3, 4, and so on. For example, for the second image, the # box is 2, and for the third image, it would be 3.

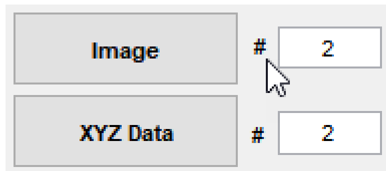


Figure 40

- The number of pixels in X and Y directions are also updated in Xn and Yn boxes.

Figure 41

- The next step is to select the data sources for feature extraction (Select Input box). One can either take Satellite Images, Point Data, or All of them together. As soon as selecting the desired data sets, the # of features box is updated in the Spatial Feature Extraction module.

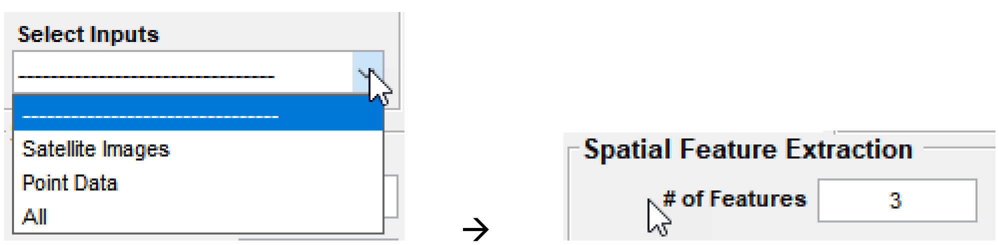


Figure 42

- Users can reduce the dimensionality by changing the value of the Dimensionality box. For example the value of 3 means no dimensionality reduction; that is, for 3 features, the output of feature extraction is the same.

Figure 43

- ICA methods also need two other parameters: ICA Convergence and ICA Max Iteration. ICA Convergence, by default, is set to 1e-13. If faster results are needed, you can reduce it to 1e-6 to 1e-5 at the expense of lesser source separation accuracy. One can also increase the number of ICA maximum iteration for better convergence to eliminate the local minima traps during optimization. By default, the number of Max Iterations is set to 100 iterations.

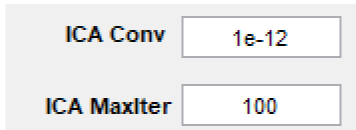


Figure 44

- The next step is to run the PCA (variance maximization), kICA (kurtosis maximization), and nICA (negentropy maximization) algorithms.

Figure 45

- After each run, a message appears that ensures the algorithm is successfully deployed (*PCA Completed, kICA Completed, nICA Completed*).

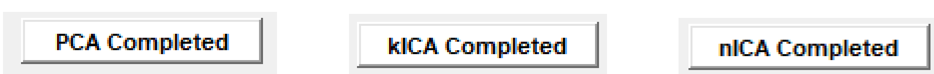


Figure 46

- For kICA and nICA methods, the best convergence error is also presented in a small box ICA Error.

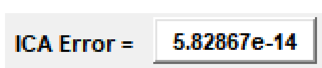


Figure 47

- A Figure window also automatically pops up that shows the ICA algorithms' performances both in linear and logarithmic plots. The logarithmic scale helps users follow the fluctuations of optimization in small ICA errors that are not visible on a linear scale.

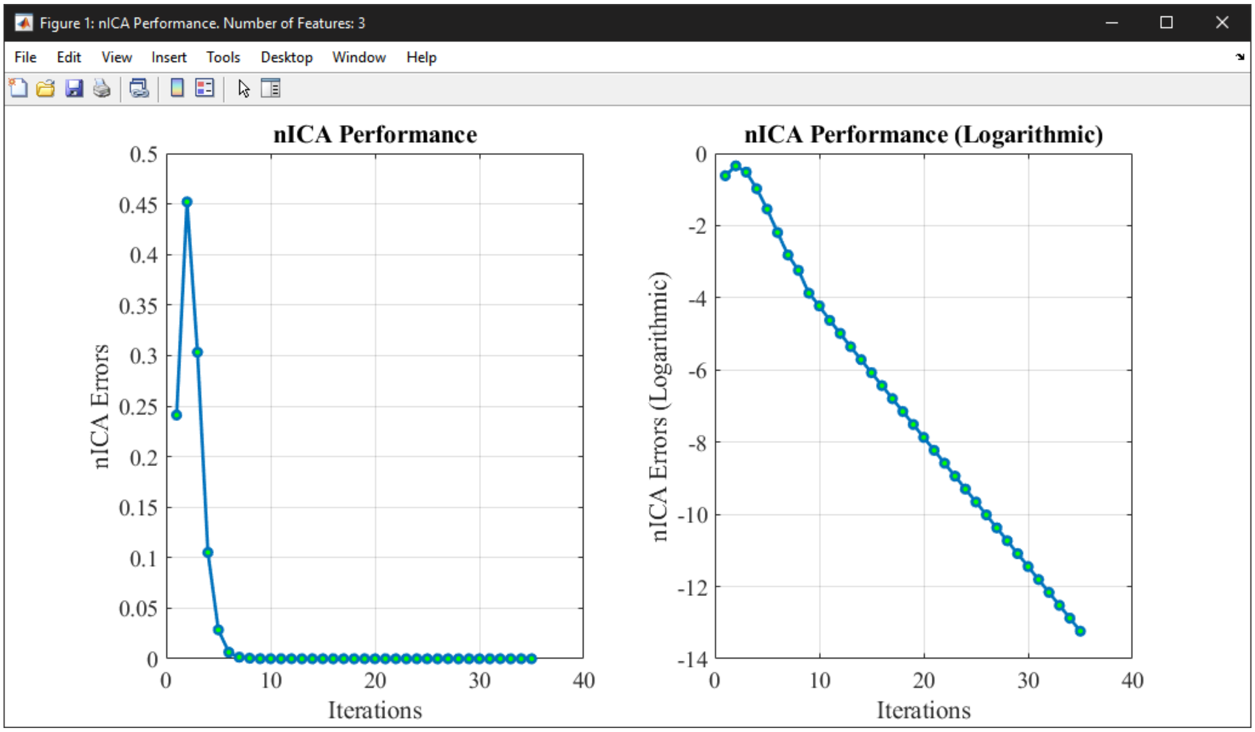


Figure 48

- Now we can see the results in the *Plot* module of the program (*Plot Results* box). We can display the input satellite images, separate RGB channels, and point data sets, together with PCA, kICA, and nICA results.

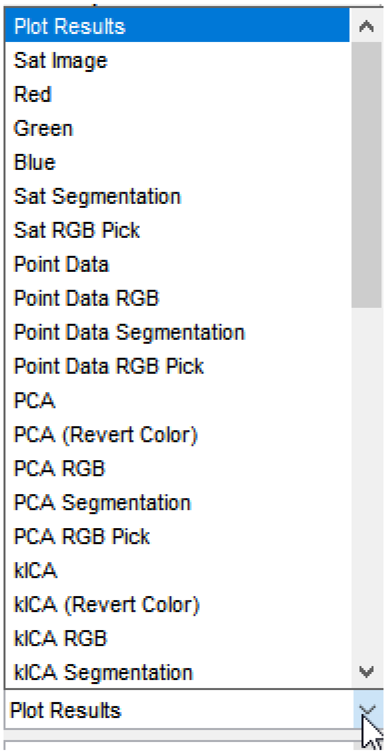


Figure 49

- For example, to display PCA results, we can select the *PCA* in the *Plot Results* box to visualize all the extracted features. *PCA (Revert Color)* shows the results with opposite polarities.

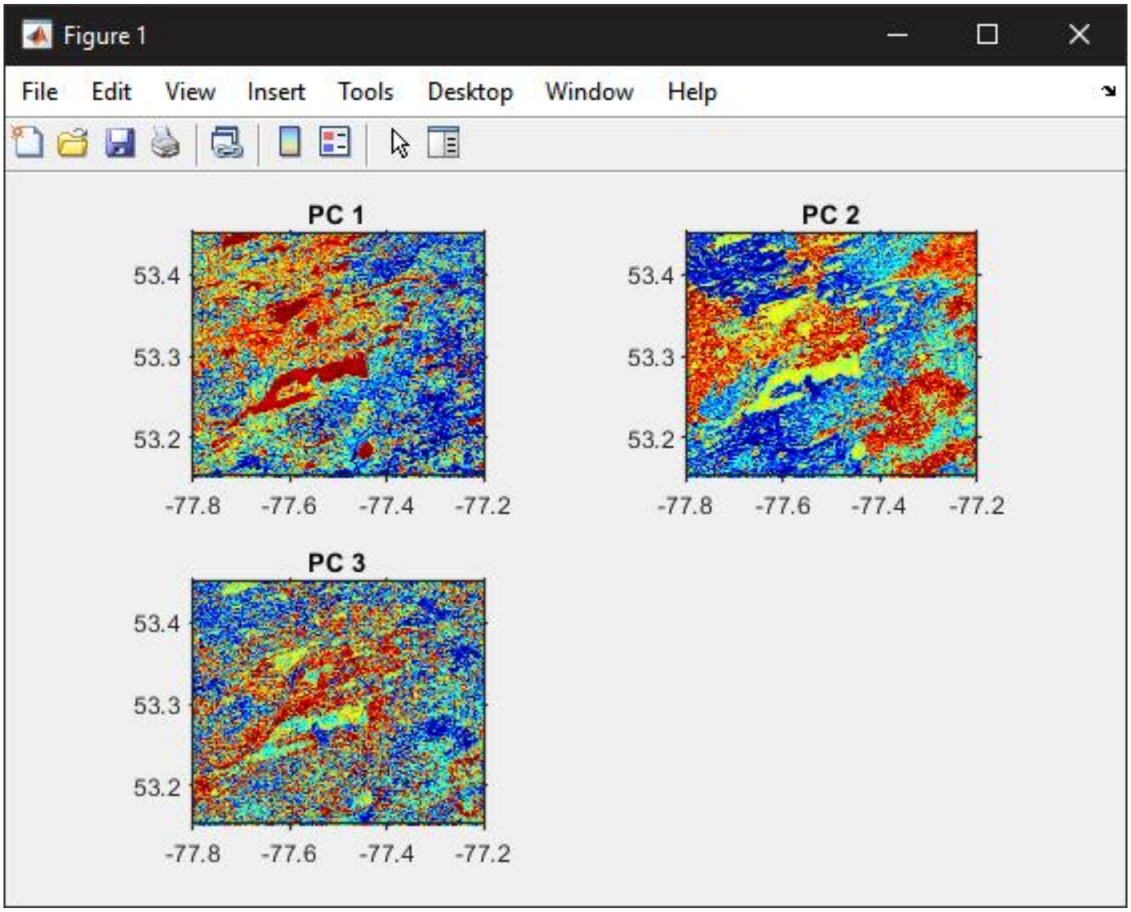


Figure 50

- We can also select the *PCA RGB* to compile a colored image out of three manually selected set of extracted features with the PCA method. To control which extract features are used for image compilation, you can set the label number of desired features in the *Features to RGB* box. For example, in nICA feature extraction, the 1, 2, 3 values mean the features 1, 2, and 3 are used for image compilation. You can also change the polarity of the features since PCA/ICA methods distort the polarities randomly. For example, you set -2, 1, 3 values in the *Features to RGB* box.

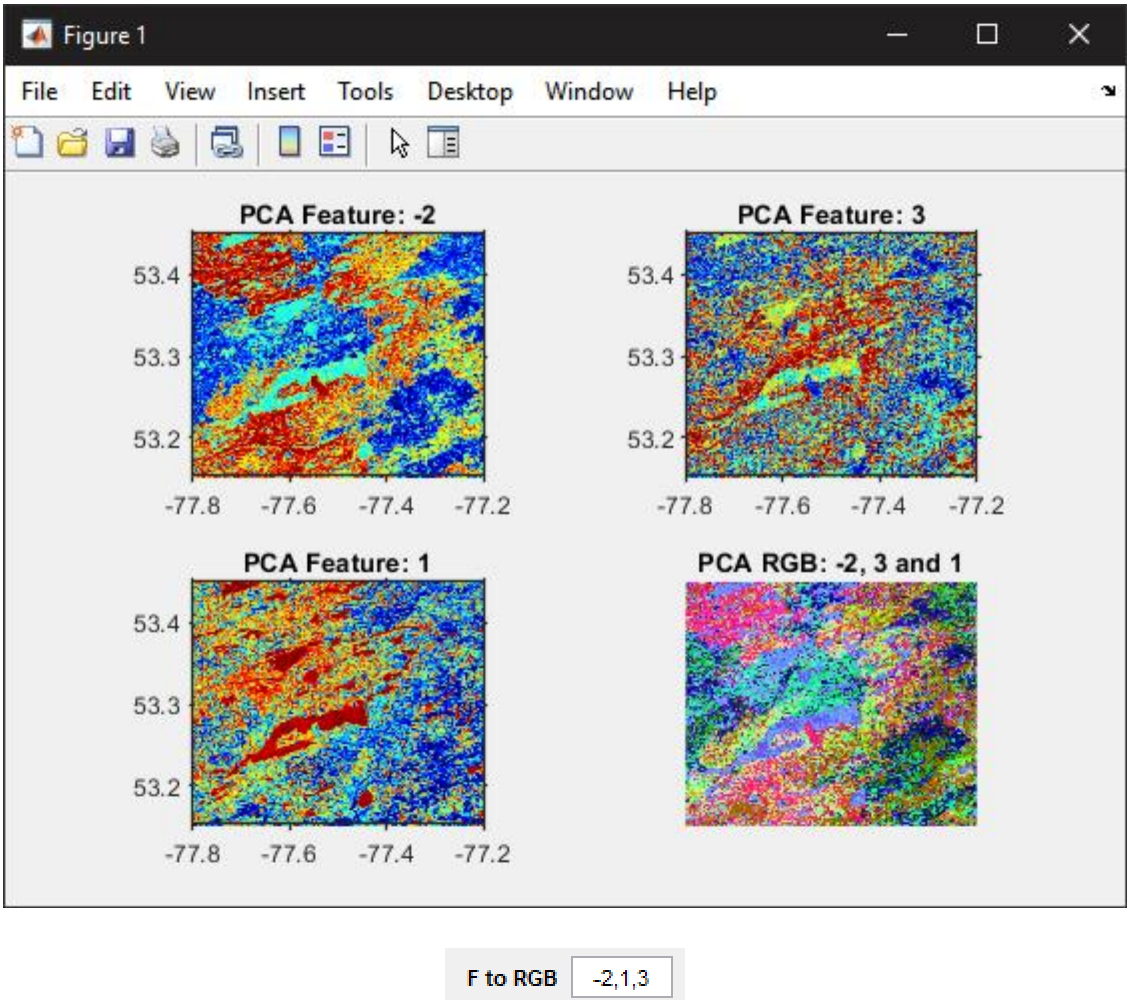


Figure 51

- *PCA/kICA/nICA RGB Pick* also displays the compiled image in a larger plot for better visualization.

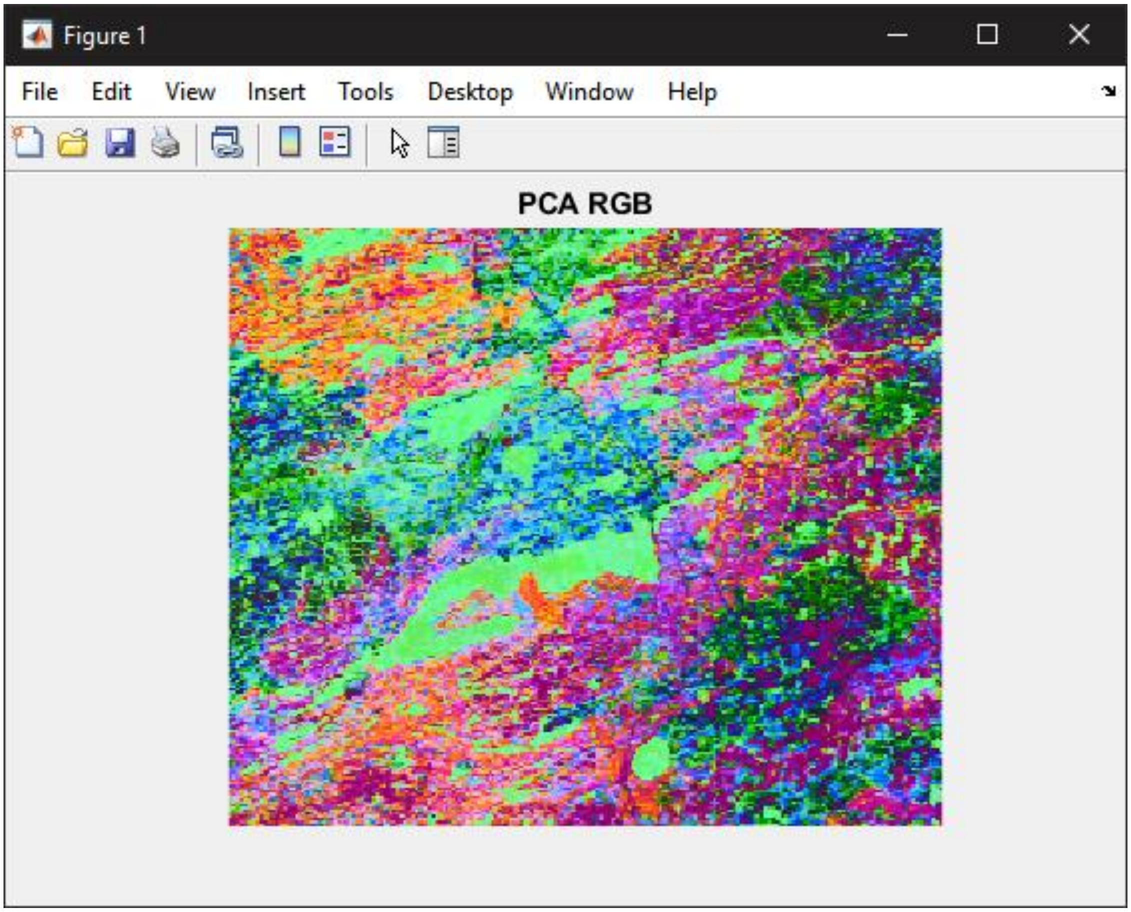


Figure 52

- For the application of *Pick Color* and Image Segmentation, please refer to sections 9 and 10.

8. Spectral feature extraction in SFES2D

Spectral feature extraction refers to the application of 2D continuous wavelet transform combine with source separation algorithms (PCA and ICA) to decompose and extract the frequency-dependent features. Dimensionality reduction is also possible. A standard way to perform a spectral feature extraction in SFES2D is as the following workflow:

- Read the coordinates.

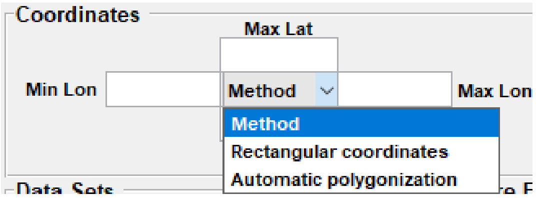


Figure 53

- Set Spacing and Data Filter values. If you do not want to smooth the data sets, just insert zero in the box any value greater than 0 will smooth the data sets proportional to the magnitude of the Data Filter.

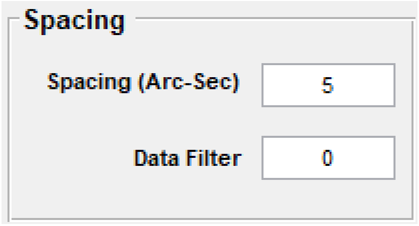


Figure 54

- Push Image/XYZ Data buttons and select the desired data sets. Users can also apply the outputs from the spatial feature extraction stage as the input of the spectral feature extraction.

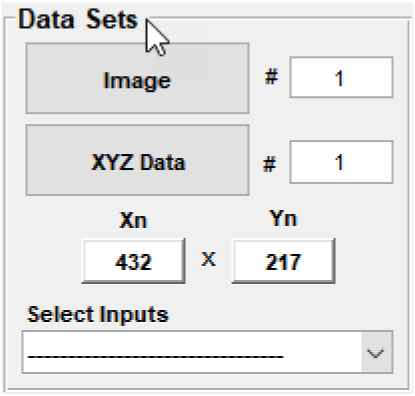


Figure 55

- In the *Spectral Feature Extraction* module, set the *Number of Scales*, *Scale Dilation*, and *Number of Angles*. For example, *Number of Scales = 3*, *Scale Dilation = 2*, CWT Filter = 1, and *Number of Angles = 8,* means that the mother wavelet expands three times at scales 1, 3, and 5 (scale dilation is the interval between scales) in 8 directions (0°, 22°.5, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°), with smoothness filter of 1 (CWT Filter = 1) that signifies that the result wavelet features are smoothed in each scale proportional to the scale (for example here for scale 1, smoothness is 1, and for scale 2, smoothness is 2, etc.). This filtering will help to get rid of the unwanted artifacts resulting from interpolation of wavelet coefficients in larger scales where the wavelet features are intrinsically smaller. CWT Filter box defines a way to set a damping factor for decomposed wavelet features. This damping factor helps to smooth the decomposed wavelet features as scales go up. Setting a very small value produces unwanted interpolation artifacts on the wavelet features. Very high CWT Filter values also eliminate valuable features. A handy rule is to set this parameter equal to the Scale Dilation.

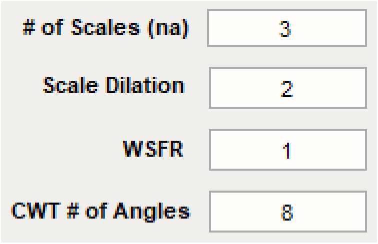


Figure 56

- Now select the desired inputs for spectral feature extraction in the CWT Inputs box. There is a list of possible options that allow users to pick a single data set, multiple images, or the

PCA/ICA results from spatial feature extraction. As soon as pushing the Merge button, with selection of the desired CWT input, the program calculates the respective scales and angles for equal intervals and updates the *Scales* and *Angles* boxes. Users are now able to manually modify the scales and angles to suit their specific workflow. The parameter β is also related to the Order of Rotational Symmetry (ORS), denoting the number of times the wavelet can be rotated by a certain angle and still maintain its original shape. This serves as an index of the wavelet's rotational invariance or symmetry. In most cases, the order of rotational symmetry of the wavelet is two, and it suffices to rotate the mother wavelet in a symmetry range between zero to 180°. In cases where the order of symmetry is four (specific cases with derivative of gaussian wavelets when m = n), one only needs to rotate the mother wavelet in a symmetry range between zero to 90°.

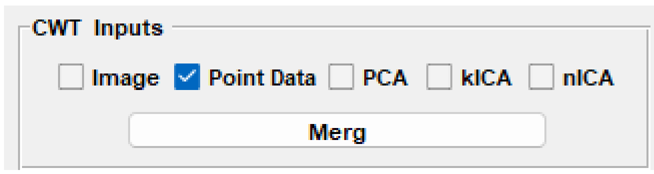


Figure 57

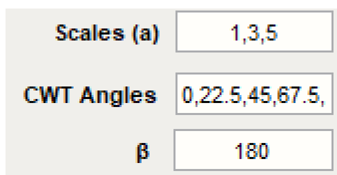


Figure 58

- Then, select the type of mother wavelet. A series of different symmetric/asymmetric mother wavelets are presented here. If you choose a symmetric mother wavelet, you should use a single direction for wavelet movement (*Number of Angles* = 1). You can modify the direction in the Angles box.

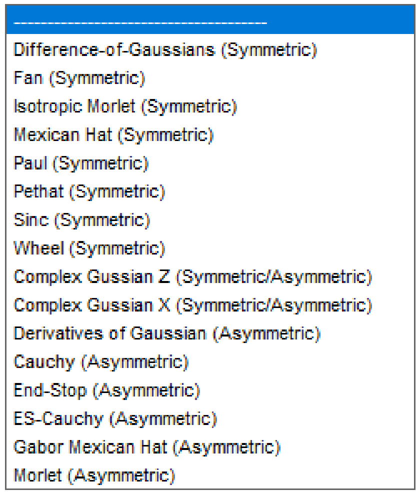


Figure 59

- For the Derivate of Gaussian wavelet, we need to define the order of differentiations in X and Y directions (n and m). We can also define whether to "change the order by scales" or not. If we select it, the algorithm tries to use high orders of differentiations in lower scales to focus more on finer edges in lower scales.

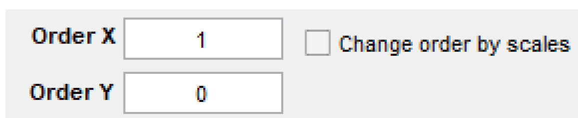


Figure 60

- For Complex Gaussian Z and Complex Gaussian X wavelets, we also need to define a similar set of parameters like the Order, the global standard deviation (Sigma), and the standard deviations in the in X and Y directions (SigmaX and SigmaY). When Sigma = SigmaX = SigmaY, the wavelet is isotropic (symmetric).

These wavelets compute the 2D X and Z order derivative of Gaussian.

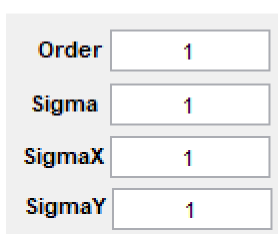


Figure 61

- The number of CWT Features (# of CWT Fs) is automatically calculated based on the number of scales and angles we set before. We can also define how to reduce the dimensionality in the “Dr to” box. Spectral ICA (S-ICA) methods need two other parameters: S-ICA Convergence (S-ICA Conv) and S-ICA Max Iteration (S-ICA MaxIter). ICA Convergence, by default, is set to 10^{-13} (1e-13). If faster results are needed, you can reduce it to 10^{-6} or 10^{-5} (1e-6 or 1e-5) at the expense of lesser source separation accuracy. One can also increase the number of ICA maximum iteration for better convergence to eliminate the local minima traps during optimization. By default, the number of Max Iterations is set to 100 iterations.

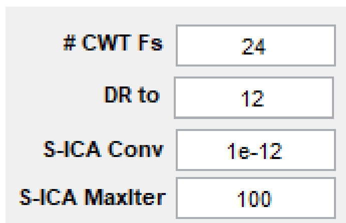


Figure 62

- Now we can use a continuous wavelet transform algorithm (*CWT* button). Now, we can separate the decomposed wavelet features to reduce the frequency-dependent overlaps. The program uses PCA (variance maximization), kICA (kurtosis maximization), and nICA (negentropy maximization) algorithms for spectral feature extraction (S-PCA, S-kICA, and S-nICA).

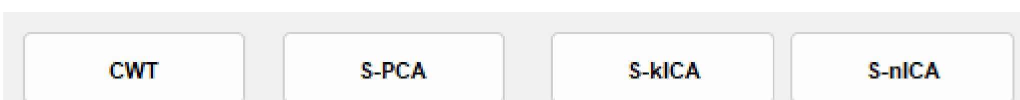


Figure 63

- After each run, a message appears that ensures the algorithm is successfully deployed (*CWT Completed*, *PCA Completed*, *kICA Completed*, *nICA Completed*).

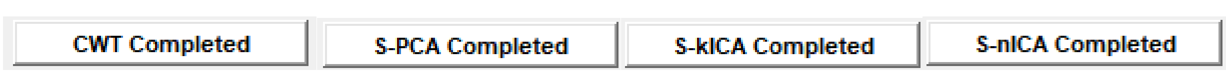


Figure 64

- For S-kICA and S-nICA methods, the best convergence error is also presented in a small box *ICA Error*.

Figure 65

- A Figure window also automatically pops up, showing the ICA algorithms' performances both in linear and logarithmic plots. The logarithmic scale helps users follow the fluctuations of optimization in small ICA errors that are not visible on a linear scale.

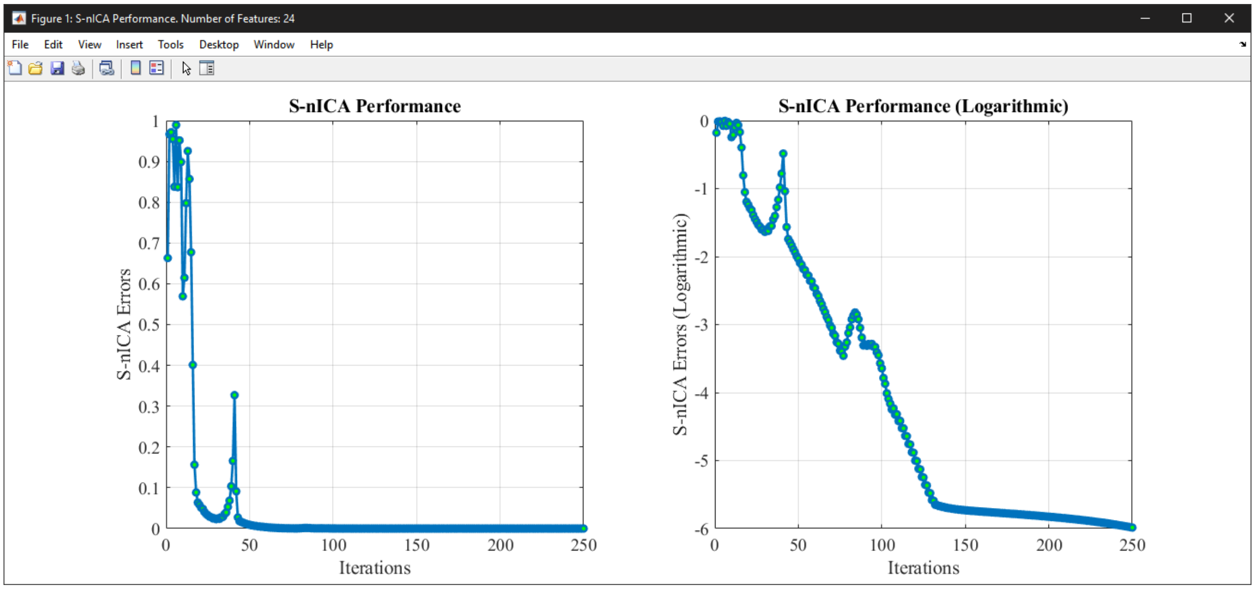


Figure 66

- Now we can see are results in the *Plot* module of the program in the *Plot Results* box. Here we can display the CWT, S-PCA, S-kICA and S-nICA results.

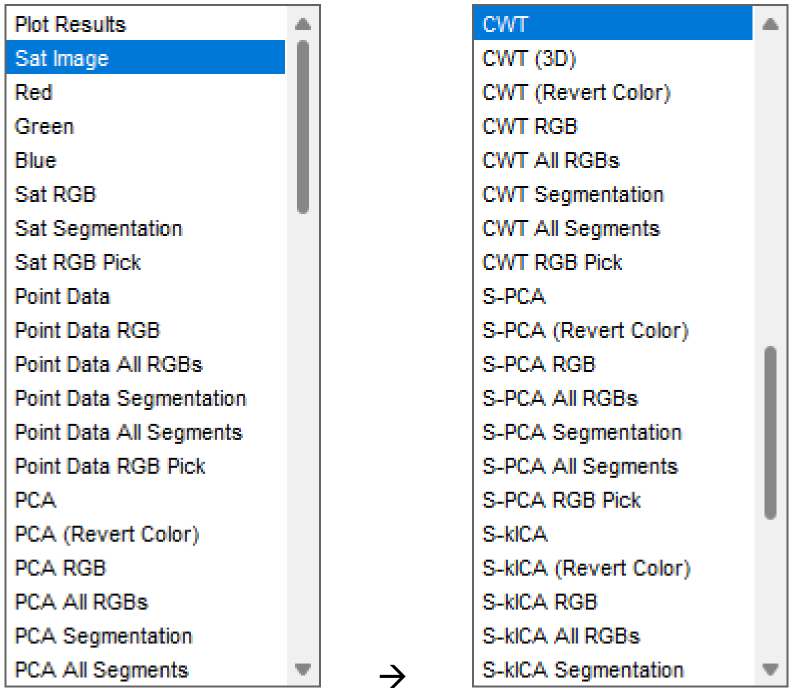


Figure 67

- For example, to display S-PCA results, we can select the *S-PCA* in the *Plot Results* box to visualize all the extracted features. We can also choose the *S-PCA RGB* to compile a colored image out of three manually selected set of extracted spectral features with the *S-PCA* method. To control which extracted features are used for image compilation, you can set the label number of desired features in the *Features to RGB* box. For example, in S-PCA feature extraction, the 1, 2, 3 values mean the features 1, 2, and 3 are used for image compilation. You can also change the polarity of the features since PCA/ICA methods distort the polarities randomly. For example, you can set -2, 1, 3 values in the *Features to RGB* box.
- *S-PCA/S-kICA/S-nICA RGB Pick* also displays the compiled image in a larger plot for better visualization.
- For the application of *Pick Color* and *Image Segmentation*, please refer to sections 9 and 10.

9. ROI selection in SFES2D

A color pick algorithm is developed for the selection of the region of interest (ROI) based on color intensities. The program assigns the low and high thresholds for each RGB color band with several clicks on the screen. The more you click on the specified zones, the more accurate the thresholding works. Then, the program automatically picks up the minimums and maximums from red/green/blue bands. To reduce the effect of the outliers, only a range of 5 to 95 percentiles are kept.

A standard way to perform an ROI selection in SFES2D is as the following workflow:

- Display the RGB image by clicking on the options in *Plot* → *Plot Results* that end with the suffix *RGB Pick*.

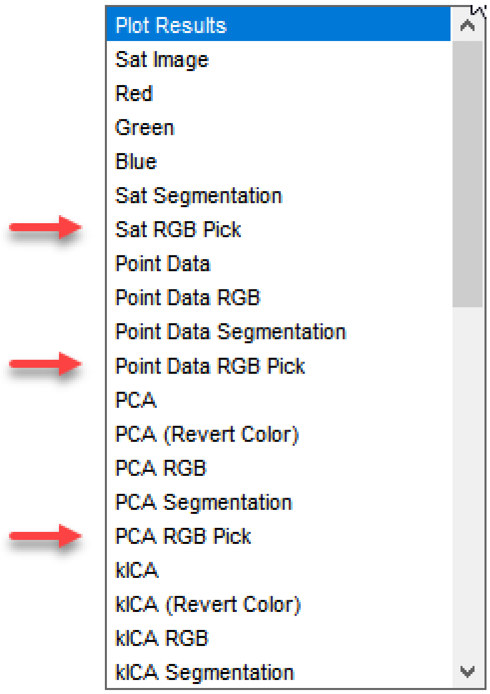


Figure 68

- Now in the *Plot module*, click on the *Pick Color* button.

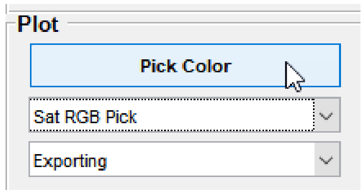


Figure 69

- Now repeatedly click on the places with interesting color signatures. The more you click, the more accurately the algorithm can detect the specified zone.

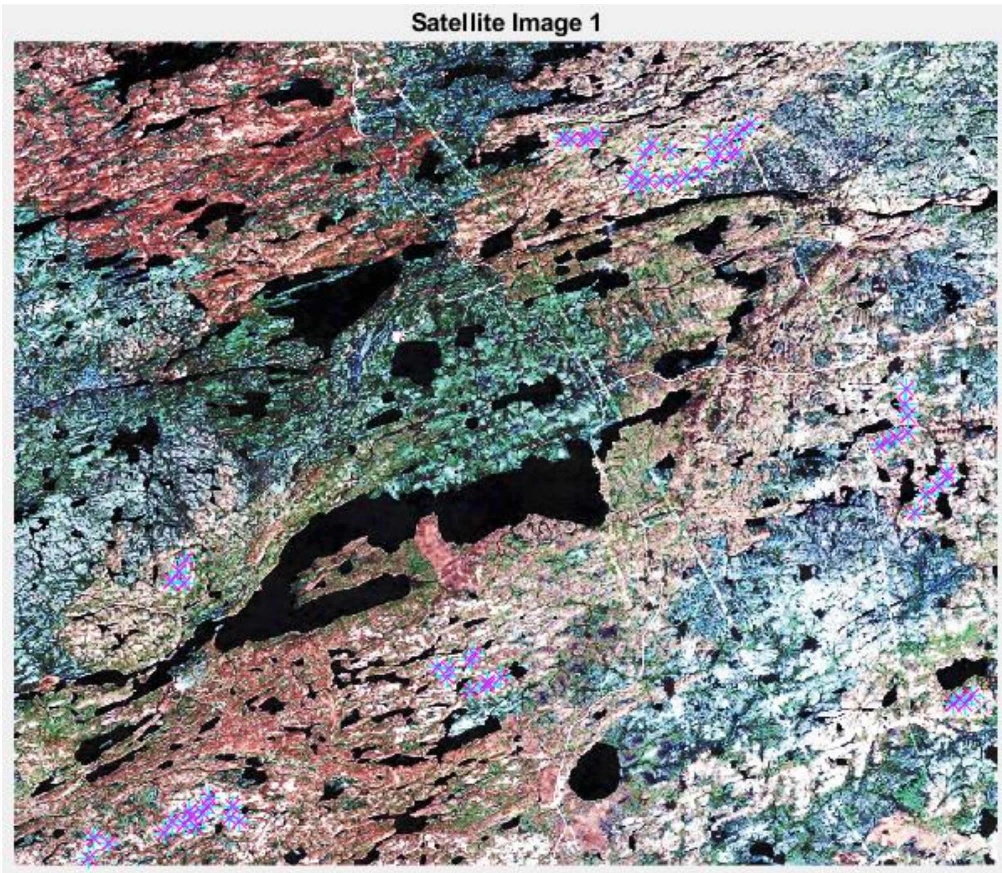


Figure 70

- After pressing the Enter, the program automatically finds all the places with similar color signatures like those you clicked and blanks out the rest.

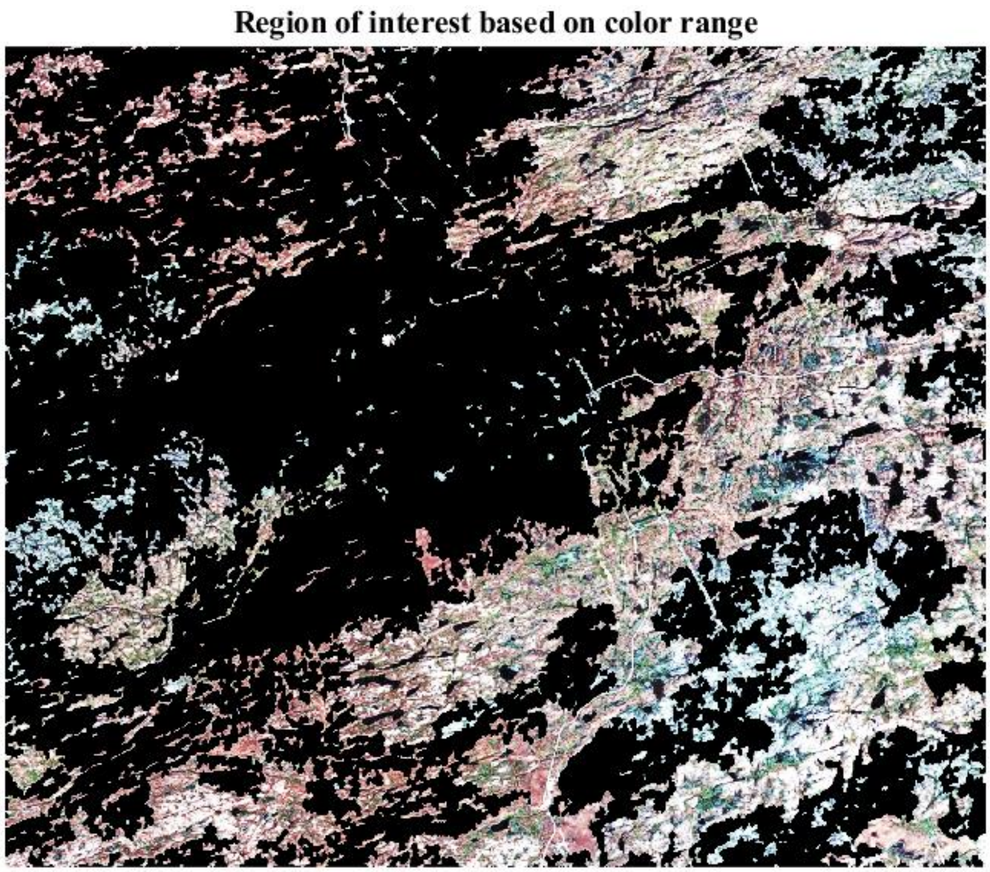


Figure 71

10. Pseudo-geological mapping by segmentation in SFES2D

SFES2D uses a fast k-mean clustering algorithm for the segmentation of color images. The segmentation helps to reduce the color space size to a manageable number of colors. The process produces a pseudo-geological map based on the spatial/spectral extracted RGB features, and eventually helps geologists detecting the hidden geological contacts and structures in geo-images. In addition, segmentation significantly reduces memory usage and speed up image analysis by focusing on relevant information.

The RGB space provides a more complex feature extraction problem where users can compile colored images from extracted features. Since every feature has two different polarities, we need to select three features out of $m = 2n$ features, where n is the number of extracted features. This is a permutation problem where $P(m,3) = m(m-1)(m-2)$ ways are possible to arrange three R/G/B features out of $m = 2n$ features. For three extracted features, we have six features, and we need to select three of them with different orders, which is 120 different ways. For four features, the number goes up to 336, and for five features, permutations are 720. As can be seen, increasing the dimensionality complicates how one can visualize feature extraction results in RGB space.

In the case of ICA-based segmentations in the RGB space, the problem gets more straightforward because the order of features is not a matter of concern in the *k-mean* segmentation. This is a combination problem where the number of ways to combine three features from a set of $m = 2n$ features is $C(m,3) = m(m-1)(m-2)/6$. However, this combination problem needs a few modifications because, although the order of features is not necessary, the order of polarities is essential in the calculation of 3D Euclidean distance in RGB space. Therefore, the problem can be reformulated in two parts:

1. Combinations of three features out of n extracted features: $C_1(n,3) = n(n-1)(n-2)/6$.
2. And the number of three combinations for all three R/G/B bands, which is $C_2 = \sum_{k=0}^{3} C(3,k) = 2^3 = 8$.

Therefore, the total number of combinations is $C_{Total} = C_1C_2 = 4n(n-1)(n-2)/3$. Table 1 shows all the possible permutations and combinations for up to 9 extracted features. As can be seen, the fraction of permutations to combinations (P/C) shows that the segmentation helps to reduce dimensionality between 15 times to 7.3 times. For very large dimensionalities ($n \rightarrow \infty$) it can be proved that the segmentation reduces the dimensionality down to six times. The SFES2D program displays all $C_2 = 8$ combinations for every C_1 combination. For very large dimensionalities, users can reduce dimensionality in ICA procedures to reduce the complexity of the problem.

Table 1. The total number of all RGB permutations (P) versus k-mean segmentations in RGB feature compilation.

n	***m***	***P***	***C_1***	***C_2***	***C***	***P/C***
3	6	120	1	8	8	15
4	8	336	4	8	32	10.5
5	10	720	10	8	80	9
6	12	1320	20	8	160	8.25
7	14	2184	35	8	280	7.8
8	16	3360	56	8	448	7.5
9	18	4896	84	8	672	7.28

A standard way to perform an ROI selection in SFES2D is as the following workflow:

- Set the # of Colors: For example, in this case, six colors are used for segmentation.

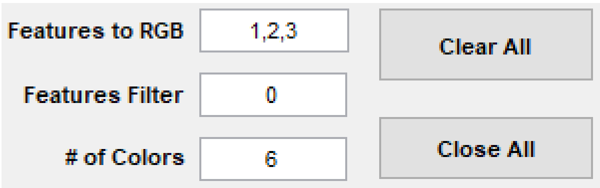


Figure 72

- Select Segmentation options in Plot → Plot Results.
- Select All Segments options in Plot → Plot Results. This option shows all eight possible combinations for the selection of three features. You can also plot all eight equivalent RGBs.

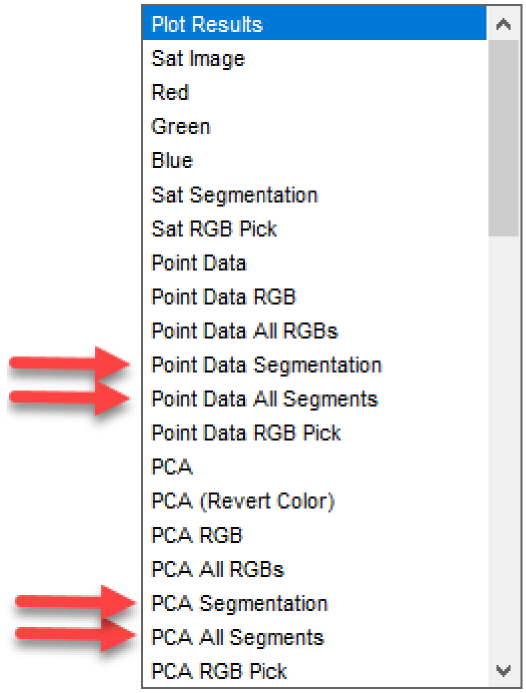
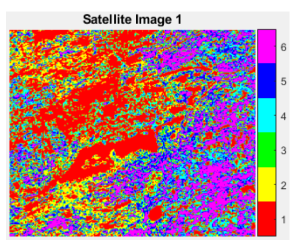
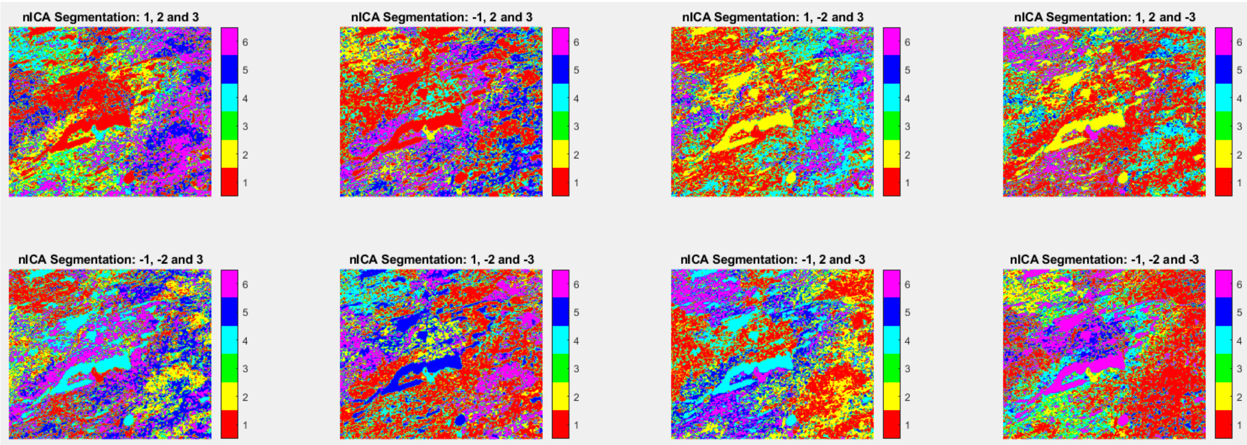


Figure 73

- An example of the segmentation on the Satellite image (from section 9) is presented below. As can be seen, spatial feature extraction has highlighted obscured structures in the segmented image.



Segmented image before source separation.



Segmented after nICA

Figure 74. An example of satellite image segmentation with SFES2D.

- An example of the segmentation on the point data sets (DEM, Gravity, and Magnetic data) is also presented below. As can be seen, spatial feature extraction has highlighted obscured structures in the segmented image.

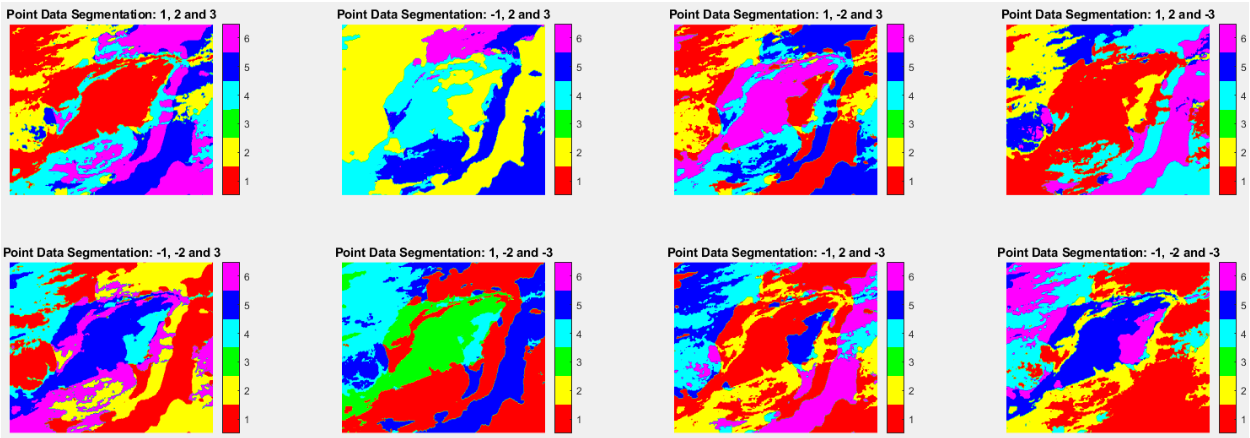


Figure 75. *Segmented point data sets (DEM, Gravity and Magnetic)*

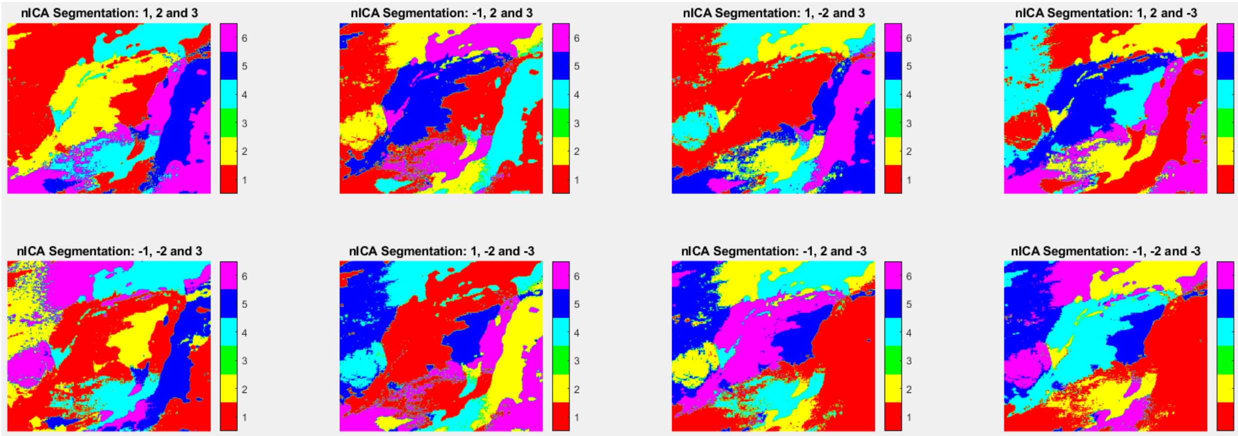


Figure 76. *Segmentation after nICA.* An example of point data sets segmentation with SFES2D.

11. Lineaments Extraction in SFES2D

In this new version (ver. 5.1) SFES2D is equipped with a novel algorithm for lineament extraction based on extraction of curvilinear patterns by wavelet-PCA hysteresis thresholding and Bayesian hyperparameter optimization (section 4.4).

11.1. Fast Lineaments Extraction with Spatial Feature Extraction

This is the fastest method of lineament extraction in cases where an estimation of parameters is already known by user, or when the fast results are more desirable than accurate ones. The algorithm can practically run on as many as possible input Images or Point Data. Here for simplicity, we only show on one single Point Data sets of Total Magnetic Field Intensities form Western Quebec, in Canada.

A standard way to perform a fast lineaments extraction in SFES2D is as the following workflow:

- Read the coordinates.

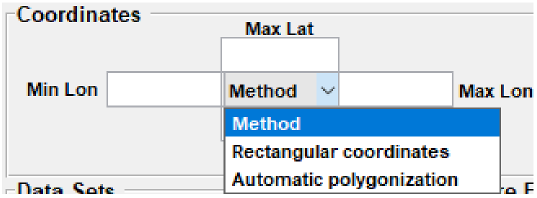


Figure 77

- Set Spacing and Data Filter values. If you do not want to smooth the data sets, just insert zero in the box any value greater than 0 will smooth the data sets proportional to the magnitude of the Data Filter.

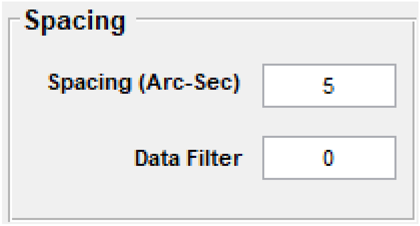


Figure 78

- Push Image/XYZ Data buttons and select the desired data sets. Users can also apply the outputs from the spatial feature extraction stage as the input of the spectral feature extraction.

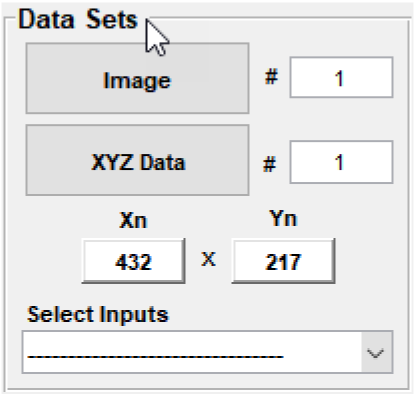


Figure 79

- Users can go on and use only their raw input images for lineaments extraction or continue with Spatial and/or Spectral feature extraction procedures (sections 7 and 8). Either way, the inputs for lineaments extraction can be selected in the box below. For example, here we only ran a fast PCA on the input data sets with an RGB satellite imagery and DEM, Gravity, and Magnetic images, comprising 6 images without dimensionality reduction.
- The next step is to set the lineament extraction parameters in the following box. “Line Res” defines the resolution of the lineament extraction output in number of pixels in the largest side of its surrounding rectangle. Larger “Line Res” results in crisper and smoother results but it takes computational cost. Here we used 320 pixels for “Line Res” parameter. “SF # of Angels” defines the number of angles for calculation of Aspect in hysteresis thresholding procedure. By default, the number is set to 18 but users can manipulate that to get better results. The width of the step filter (w) by default is often 10 percent of the largest pixel numbers. For example. Here for Line Res of 320 pixels, we used a width of w = 32. the Variability of the Step Filtering Widths (VSFW) is also by default set to 0.25. The VSFW plays a pivotal role in determining the variability of w in correlation with the complexity of the extracted spectral features. A larger value of VSFW yields a more significant w for less complex features and a reduced w for more linear features. For VSFW = 0, the values of w for each lineament extraction iteration on wavelet-PCA features remain constant. This dynamic adjustment engenders a variable measure for capturing the curvilinearity of each spectral feature, dependent on its complexity.

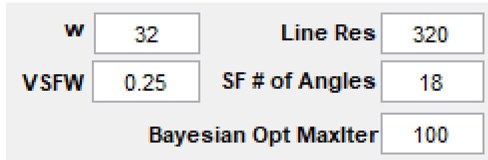


Figure 80

- To upload the digitized target faults, we use the Validation Data box.

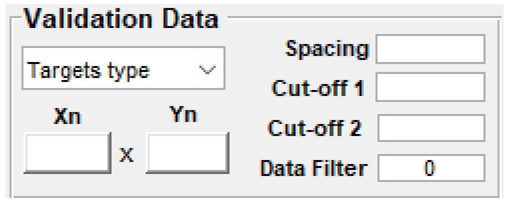


Figure 81

- We need also to define the spacing for target interpolation which is usually smaller or equal to the Data Sets spacing.

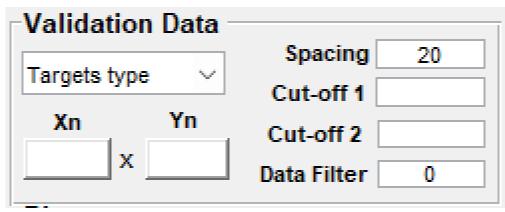


Figure 82

- There are two target types:

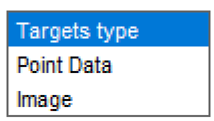


Figure 83

The targets in the form of Point Data are CSV files in XYZ format. The targets in the form of Image need also a coordinate for the georeferencing the target image. A window will open after selecting the Image to upload the coordinates for target georeferencing in a .txt format (for the Data Point form, one need no georeferencing .txt file).

- As soon as selecting Point Data or Image the value inside the Cut-off 1 box is going to be updated. Cut-off 1 determines the buffer zone around the target samples in XY plane.

Figure 84

- The default value of Cut-off 1 factor is unchangeable and is determined in a way it produces the smallest possible buffer around the data points.

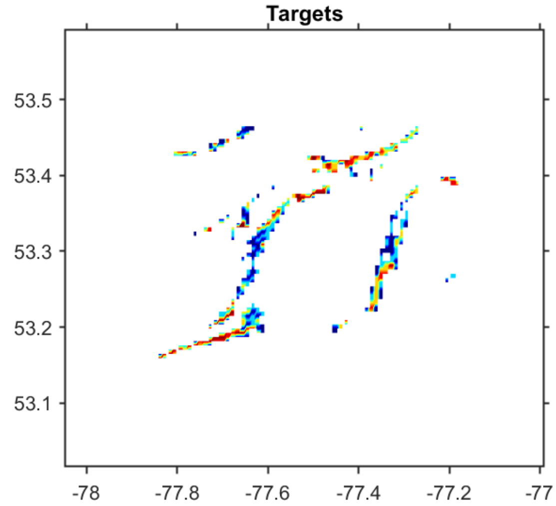
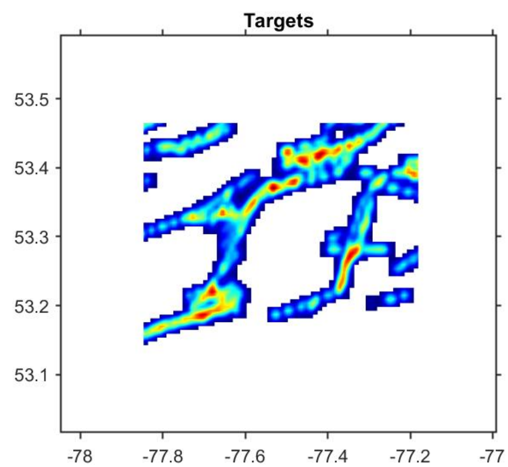
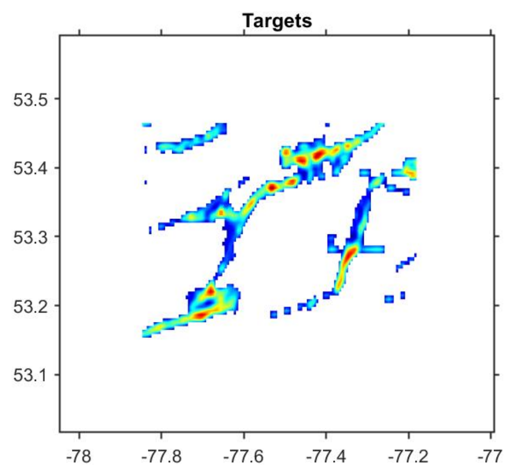


Figure 85

- To expand the buffer zone users can increase the Data Filter value. The Cut-off 2 factor also contracts the buffer zone. Manipulating these factors helps to delineate the targets in their real locations and eliminate the places without information about target property.





Data Filter = 1, Cut-off 1 = 0.84127 Data Filter = 1, Cut-off 1 = 0.84127, Cut-off 2 = 0.86

Figure 86

- Then we need to select the Input Features for lineament extraction:

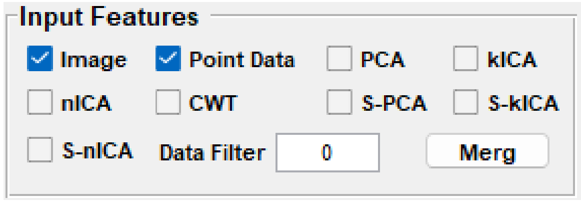


Figure 87

- Results of the lineaments extraction on raw images are as follows.

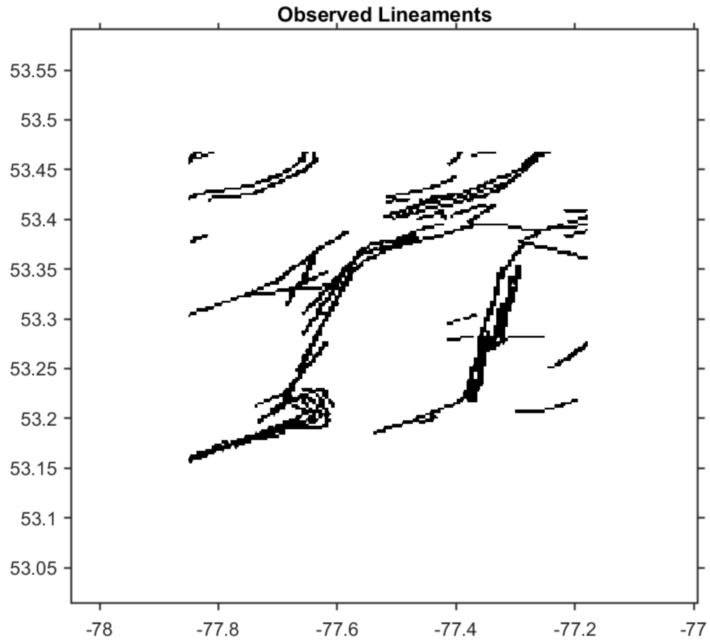


Figure 88

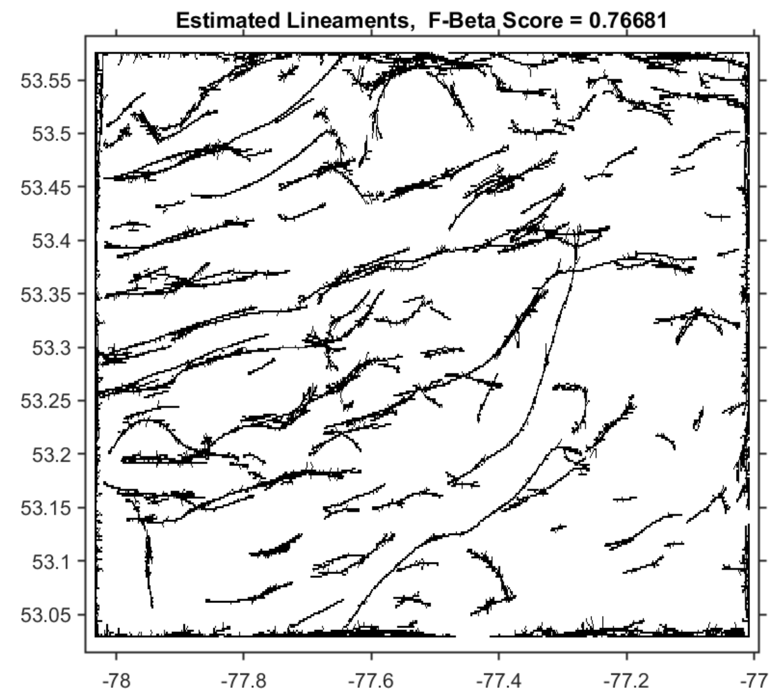


Figure 89

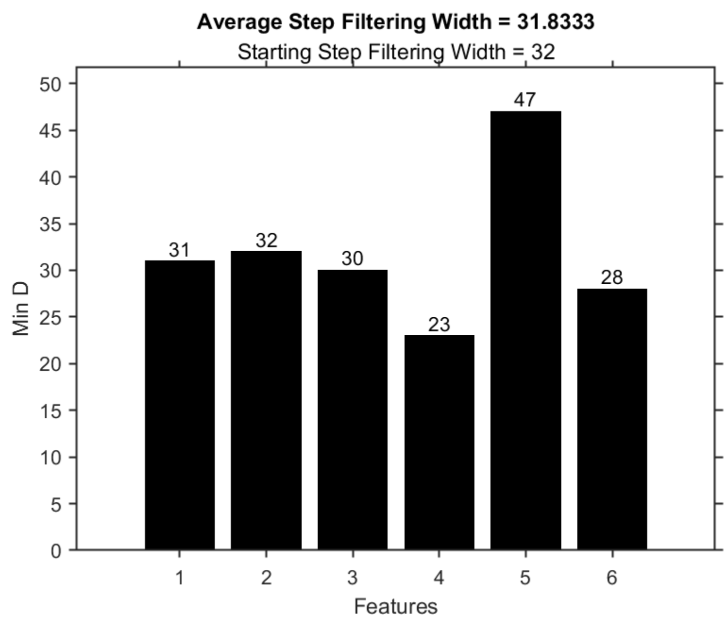


Figure 90

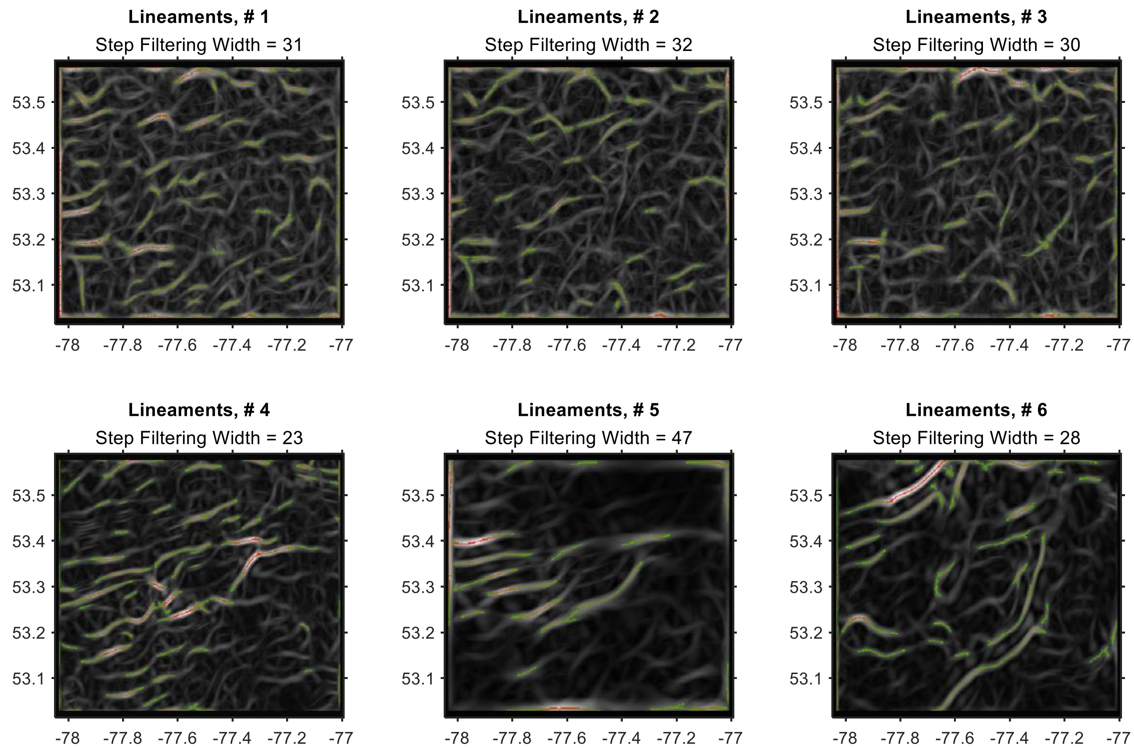


Figure 91

- And results after PCA Spatial Feature Extraction:

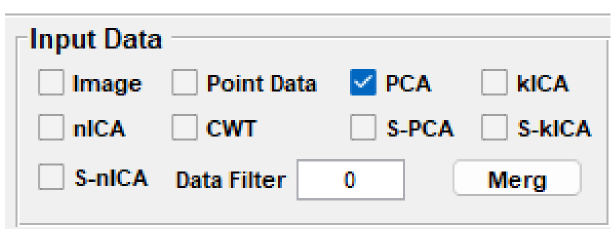


Figure 92

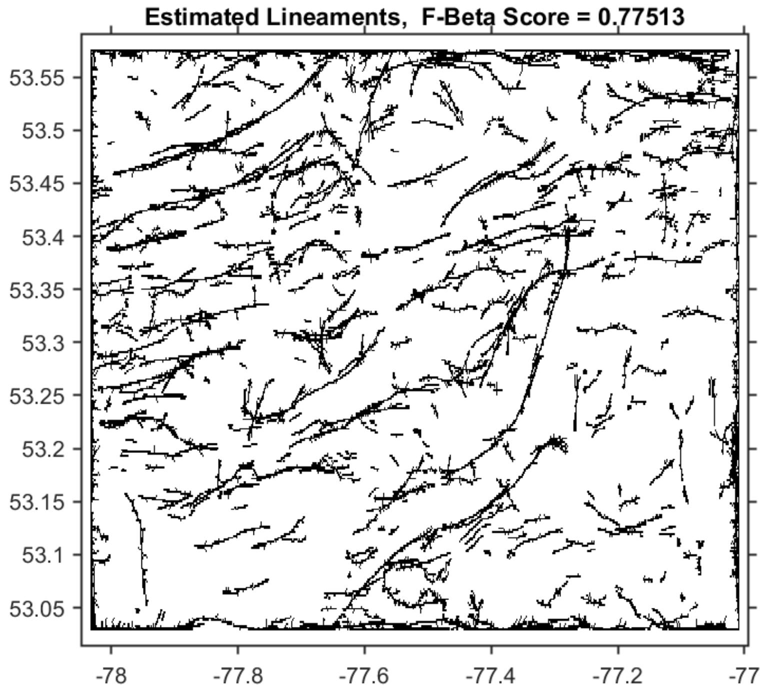


Figure 93

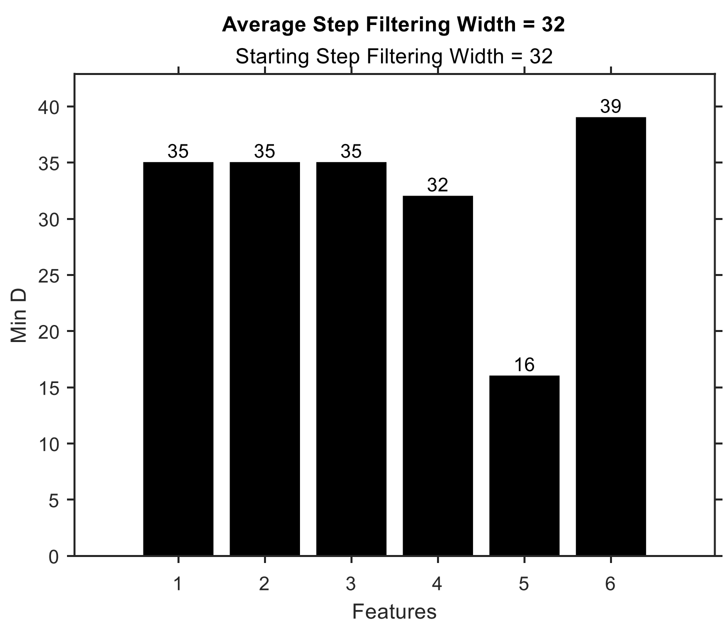


Figure 94

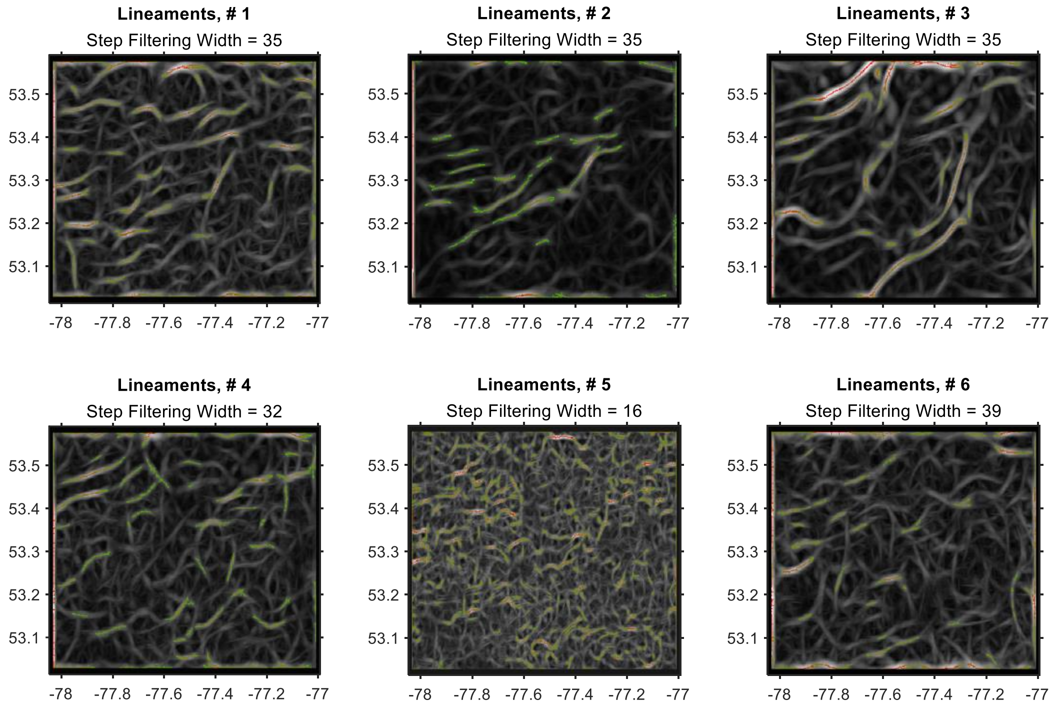


Figure 95

- And results after nICA Spatial Feature Extraction:

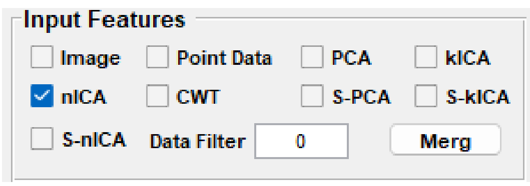


Figure 96

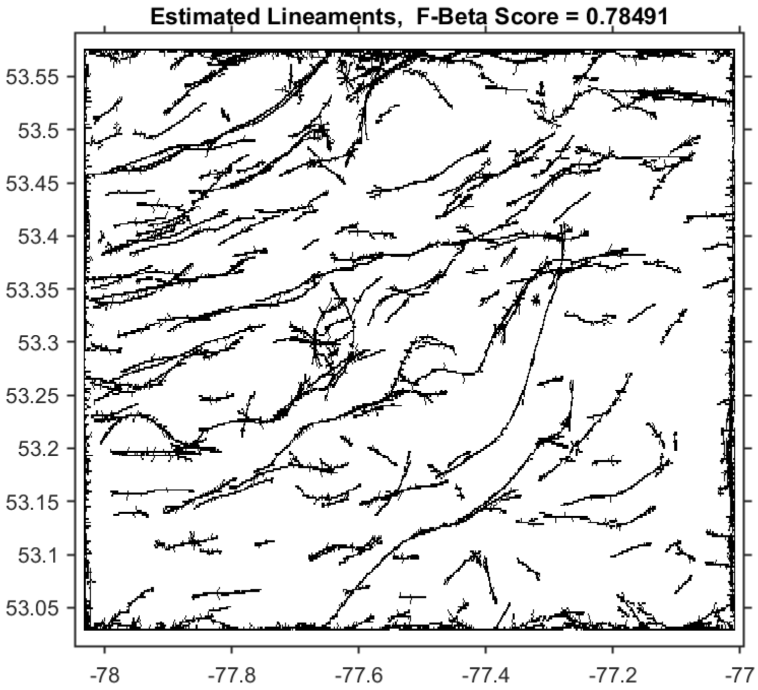


Figure 97

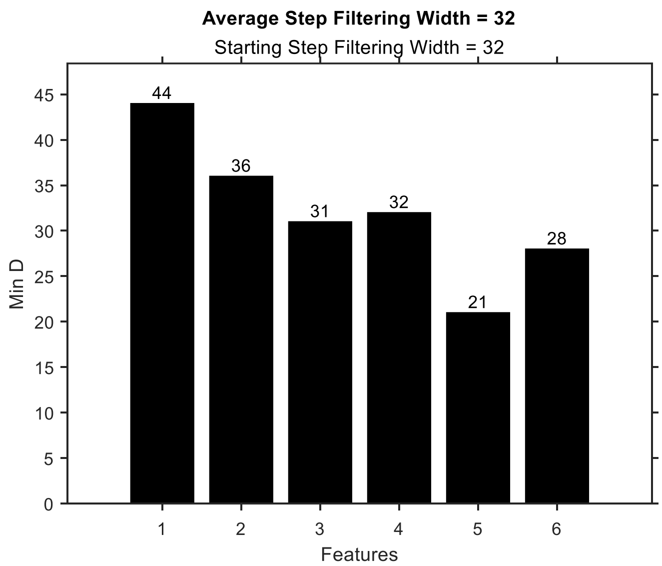


Figure 98

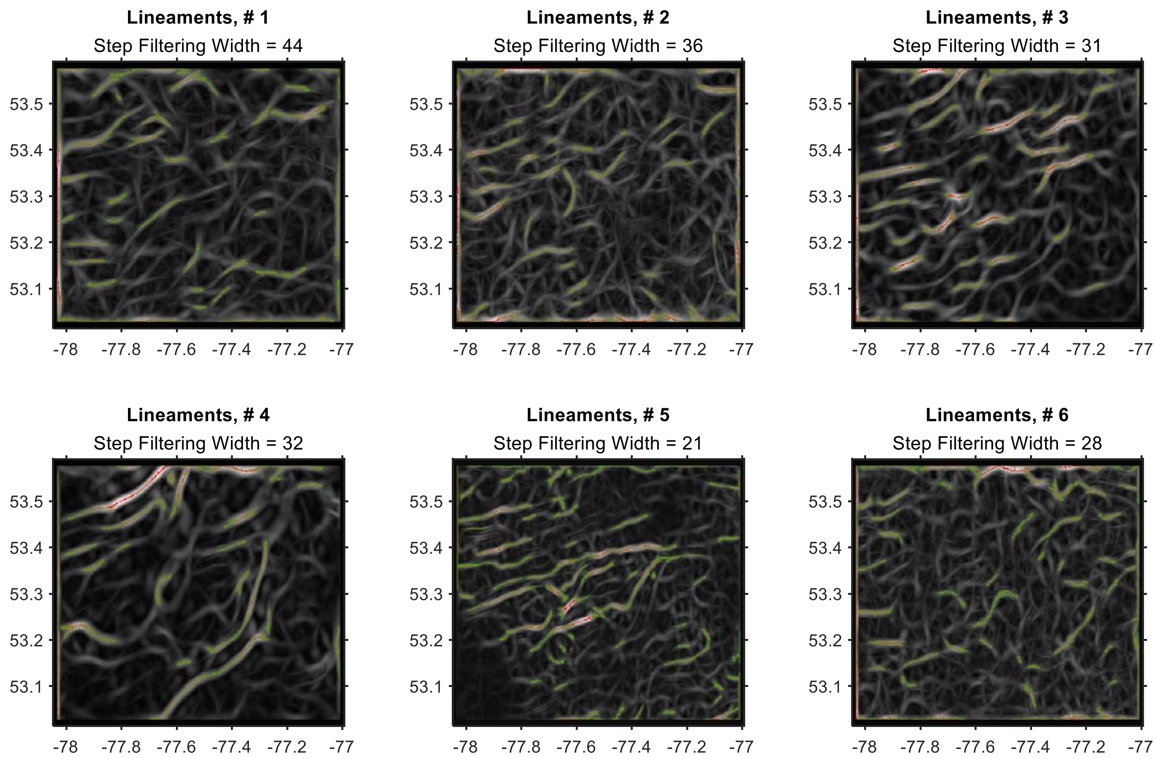


Figure 99

11.2. Fast Lineaments Extraction with Spectral Feature Extraction with DOG wavelets

- The following set of wavelet parameters are selected for lineament extraction. The number of scales is set to 3 with scale dilation of 1 and Wavelet Smoothness Filter Ratio (WSFR) of 0. Eight different directions are set for wavelet shifting those results in a = {1, 2, 3} and CWT Angles = {0°, 22°.5, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°}. Since we used derivatives of the Gaussian as mother wavelet with n = 1 and m = 0, the resulting wavelet has a β = 180° symmetry (in cases where n = m, we will have β = 90°).

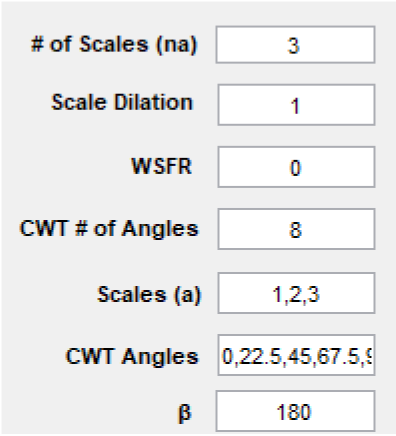


Figure 100

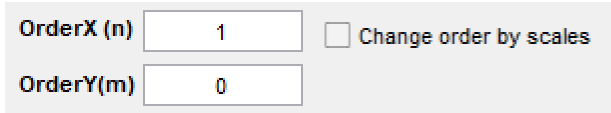


Figure 101

- Dimensionality is also reduced to 10 from 144 CWT features.

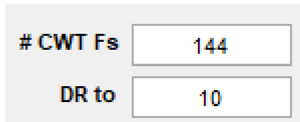


Figure 102

- This time we only use S-PCA for lineaments extraction.

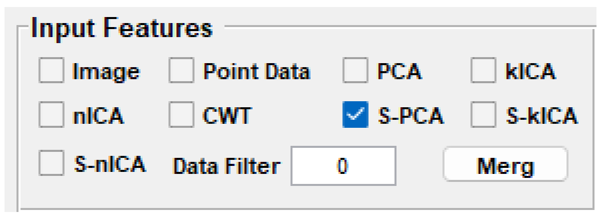


Figure 103

The results of lineaments extraction are present bellow.

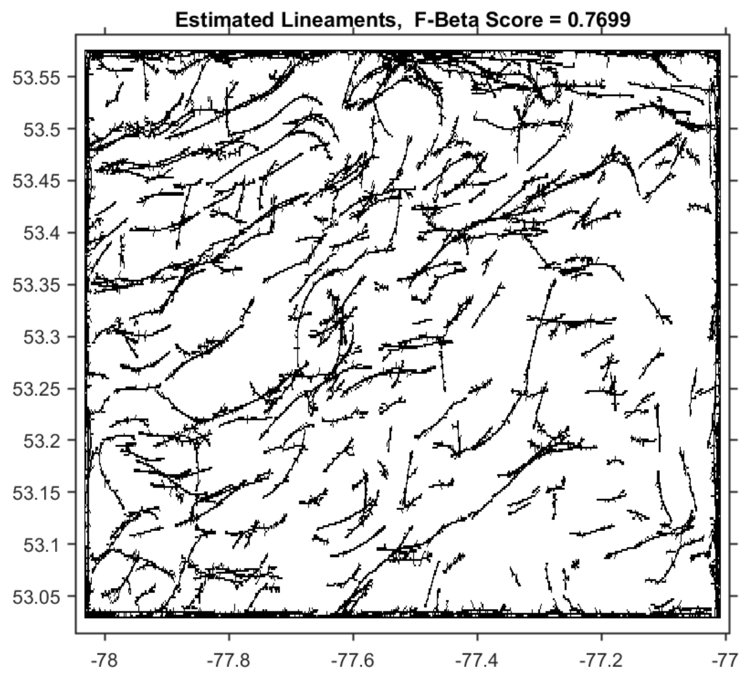


Figure 104

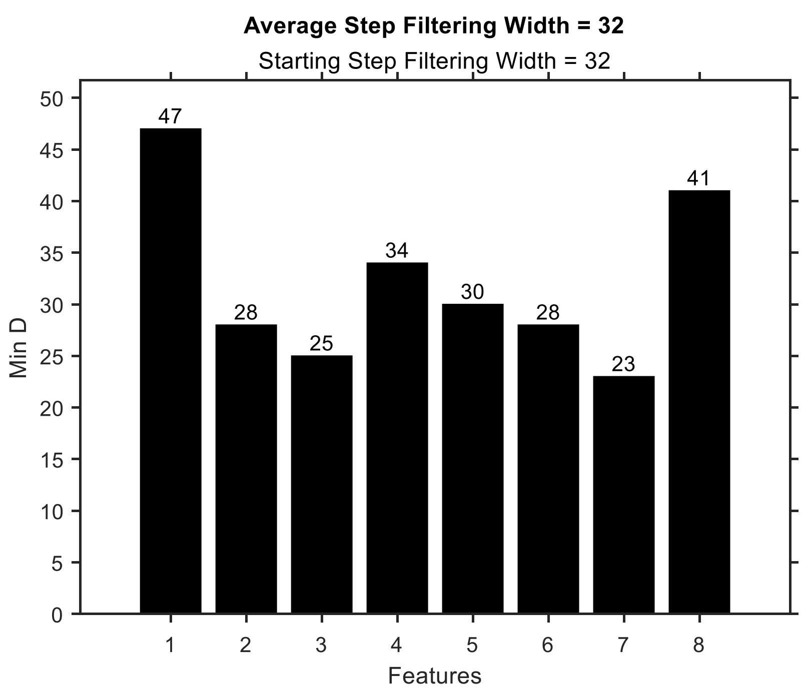


Figure 105

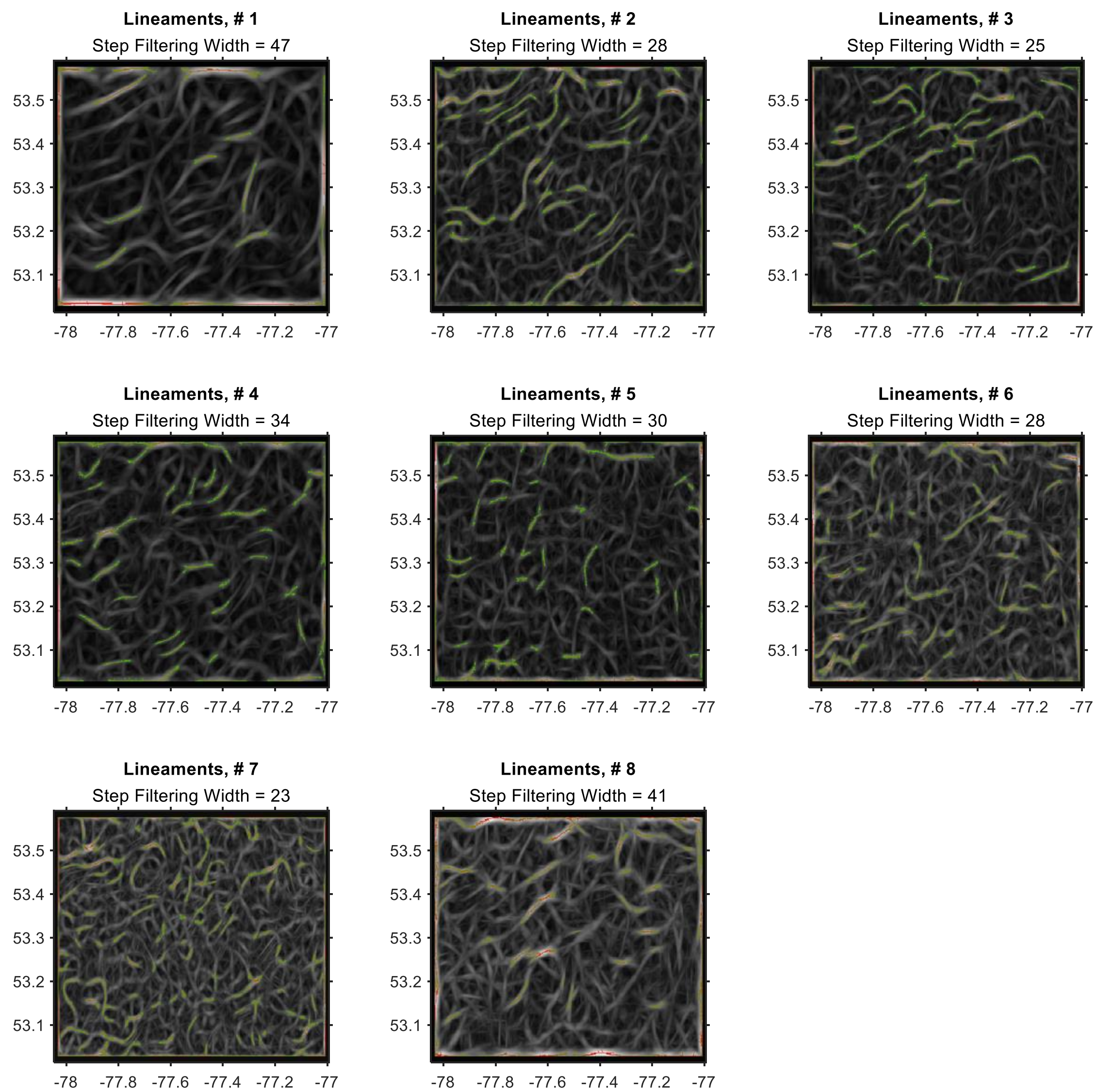


Figure 106

11.3. Lineaments extraction by DOG Wavelet-PCA feature extraction with Bayesian hyperparameter optimization

We use AutoLine Botton for Bayesian hyperparameter optimization. It is based on DOG Wavelet-PCA feature extraction.

- The following set of initial wavelet parameters are selected for lineament extraction optimization. The number of scales is set to 3 with scale dilation of 1 and Wavelet Smoothness Filter Ratio (WSFR) of 0. Eight different directions are set for wavelet shifting those results in a = {1, 2, 3} and CWT Angles = {0°, 22°.5, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°}. Since we used derivatives of the Gaussian as mother wavelet with n = 1 and m = 0, the resulting wavelet has a $\beta = 180°$ symmetry (in cases where n = m, we will have β = 90°).

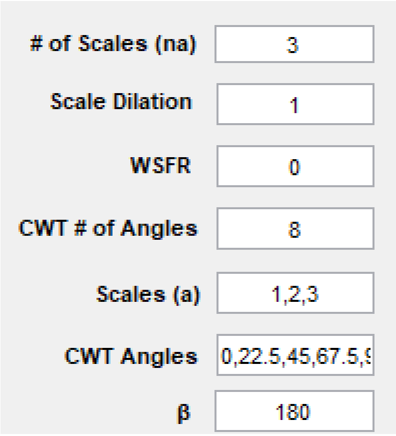


Figure 107

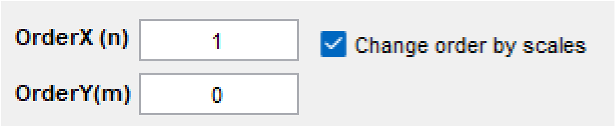


Figure 108

- For Bayesian optimization we set the n and m change by scales. In this case, the algorithm tries to find the optimal number of scales (n_a) which in turn determines the subsequent n and m values according to table 2 (in this case we should tick the Change order by scales option). The following procedure is used to associate the a to n and m:

```
for jj = 1 : N_Angles
    for jjj = 1 : N_Scales
        if jjj == N_Scales - 0
            orderx = 1;
            ordery = 0;
        elseif jjj == N_Scales -1
            orderx = 2;
            ordery = 0;
        elseif jjj == N_Scales -2
            orderx = 1;
            ordery = 1;
        elseif jjj == N_Scales -3
            orderx = 2;
            ordery = 1;
        elseif jjj == N_Scales -4
            orderx = 2;
            ordery = 2;
        end
        if orderx == ordery
            betad = 90;
        else
            betad = 180;
        end
    end
end
```

Figure 109

The Optimization is set to change the values between 1-5. For example:

For n_a = 1, we have, a = {1}, and orderx = 1; ordery = 0.

For n_a = 2, we have, a = {1, 2}, and for a = 1, we have orderx = 2, and ordery = 0, and for a = 2, we have orderx = 1; ordery = 0.

The table 2 summarize the values for all number of scales up to 5:

Table 2. Change order of differentiations and β by scales.

$n_a = 5$				$n_a = 4$				$n_a = 3$				$n_a = 2$				$n_a = 1$			
a	***m***	***n***	***β***	***a***	***m***	***n***	***β***	***a***	***m***	***n***	***β***	***a***	***m***	***n***	***β***	***a***	***m***	***n***	***β***
5	1	0	180	**4**	1	0	180	**3**	1	0	180	**2**	1	0	180	**1**	1	0	180
4	2	0	180	**3**	2	0	180	**2**	2	0	180	**1**	2	0	180				
3	1	1	90	**2**	1	1	90	**1**	1	1	90								
2	2	1	180	**1**	2	1	180												
1	2	2	90																

- Dimensionality is also reduced to 10 from 144 CWT features.

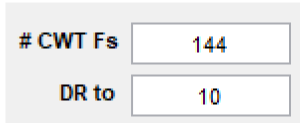


Figure 110

- This time we only use S-PCA for lineaments extraction.

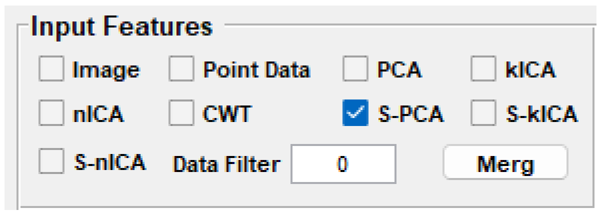


Figure 111

The results of lineaments extraction optimization are present below.

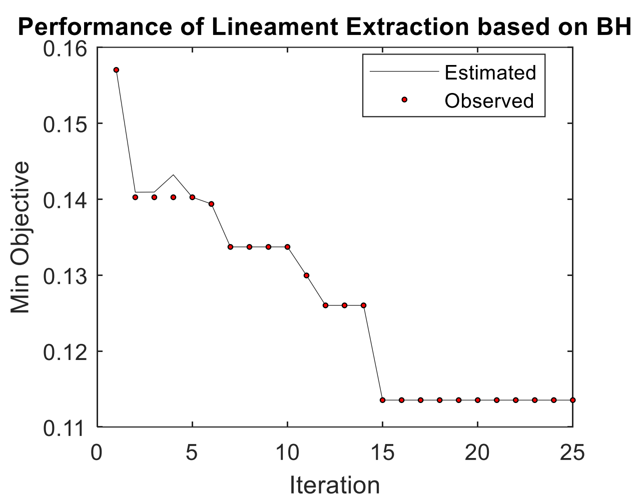


Figure 112

- The results of hyperparameters optimized with Bayesian optimization show that the values of parameters converge to a single value and sometimes into a narrow range of values. One can also use a value inside the colored box for the last 5 iterations, or a median of all those values as a reference.
-

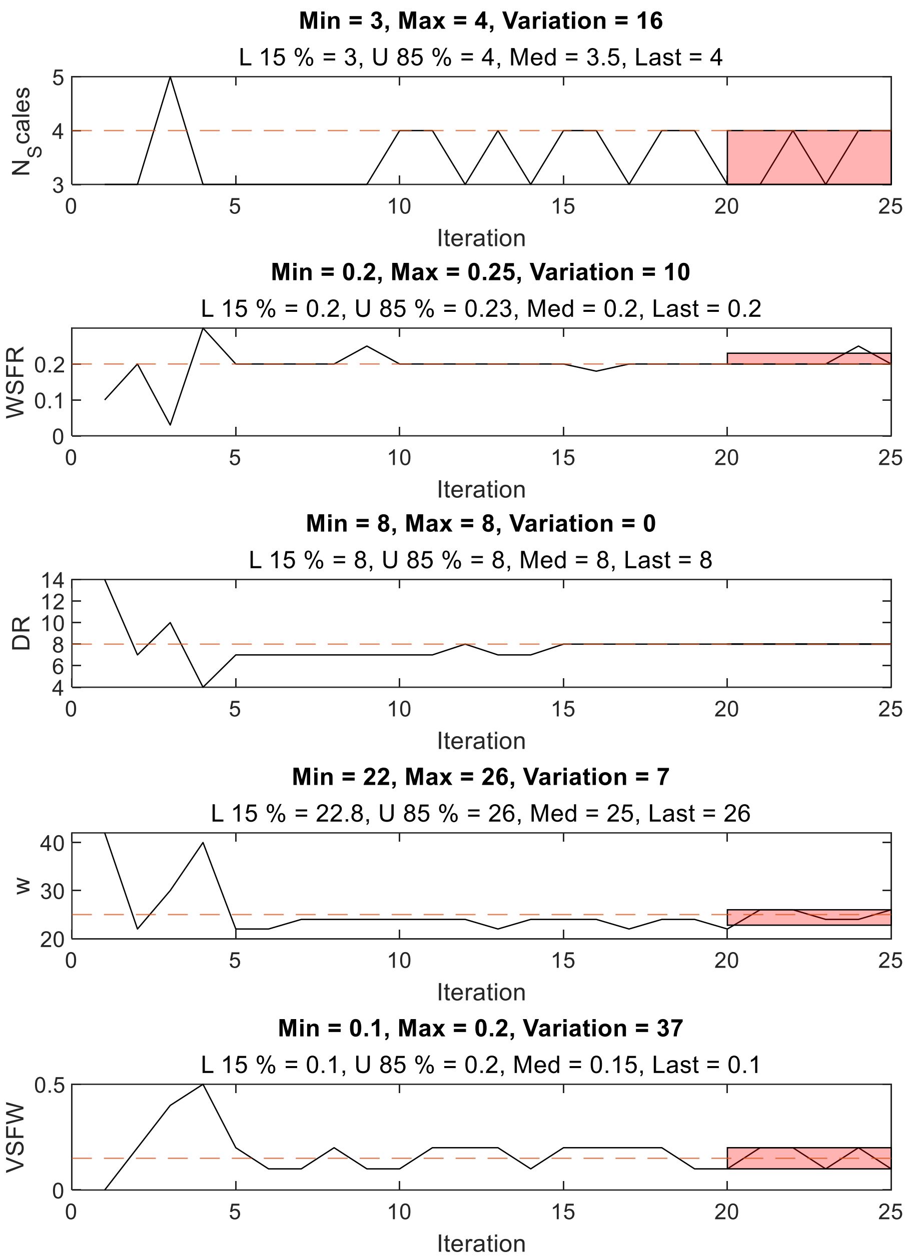


Figure 113

- As can be seen, for example the number of scales (n_a) is optimized to 4 in the last iteration. The program automatically exports the optimized median values to the program. But users can also manually change them according to the results in previous figures.
- In order to run feature extraction algorithm, once again we push the Merg button to prepare inputs for CWT calculations with new optimized parameters.

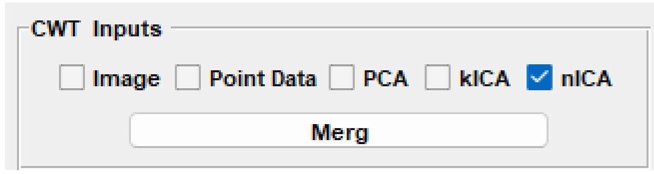


Figure 114

- then we pish the CWT button:

Figure 115

- Note that the DR to value is automatically reset to 192. So, set it back to 8 after running the CWT and before S-PCA.

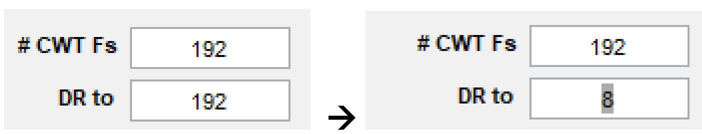


Figure 116

- Now we push the Lineaments button after selecting the S-PCA for Input Features.

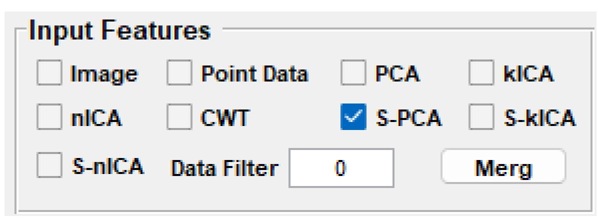


Figure 117

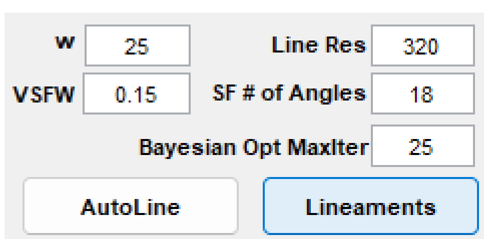


Figure 118

- The result of lineament extractions with optimized hyperparameters are presented below:

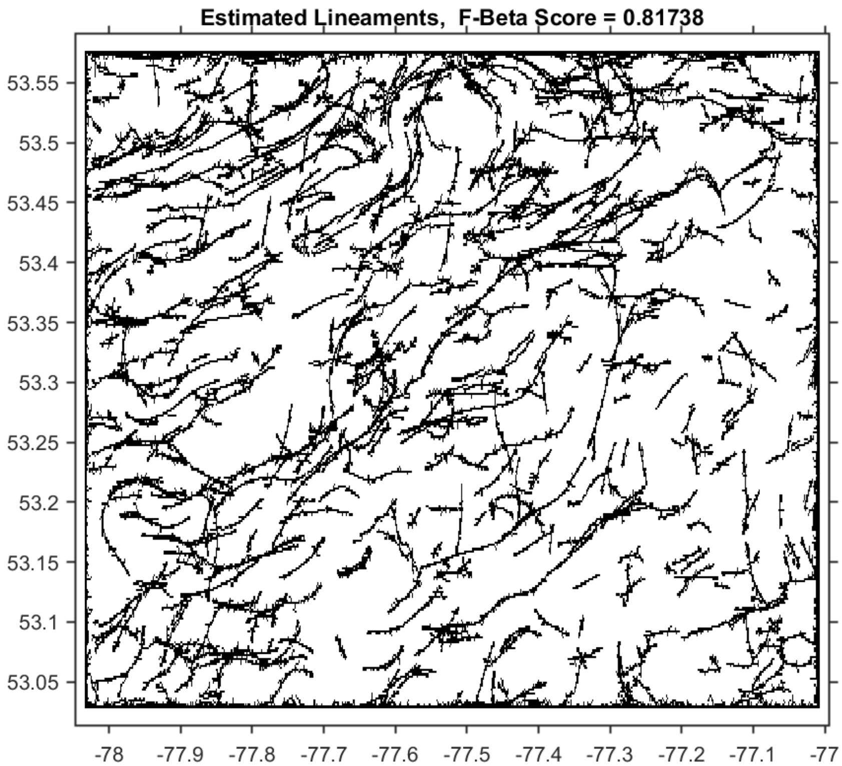


Figure 119

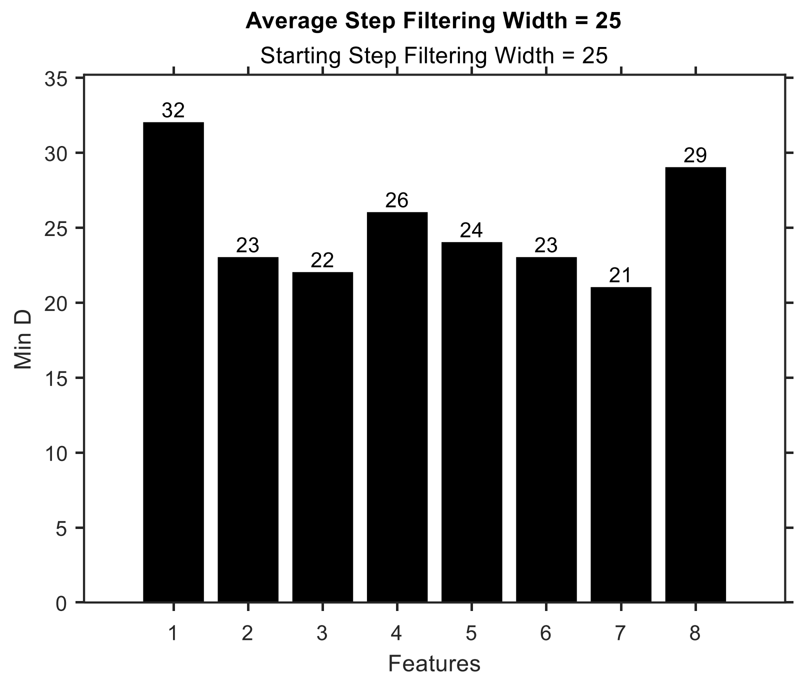

Average Step Filtering Width = 25
Starting Step Filtering Width = 25
Min D
Features
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
32
23
22
26
24
23
21
29

Figure 120

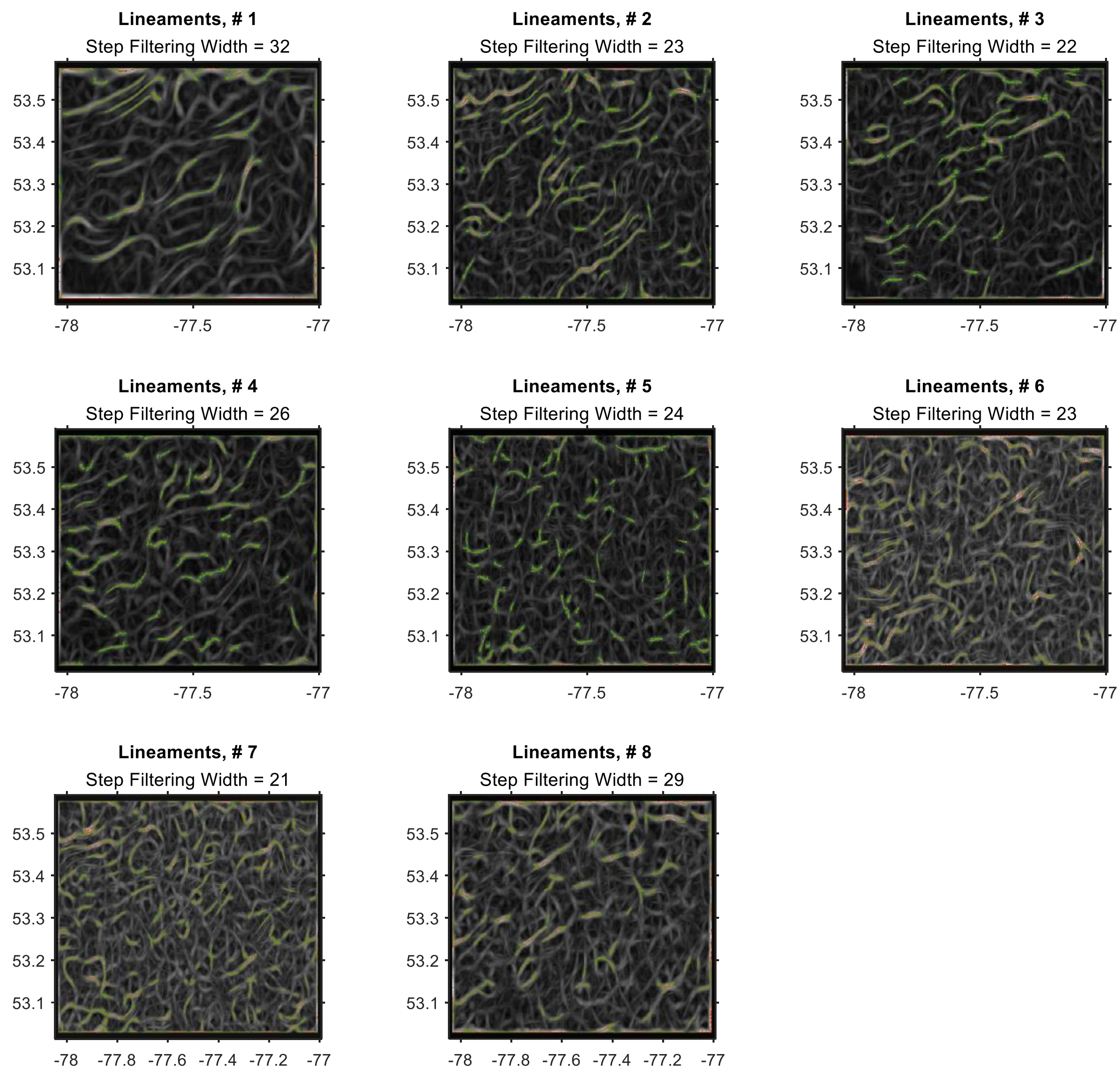

Lineaments, # 1
Step Filtering Width = 32
Lineaments, # 2
Step Filtering Width = 23
Lineaments, # 3
Step Filtering Width = 22
Lineaments, # 4
Step Filtering Width = 26
Lineaments, # 5
Step Filtering Width = 24
Lineaments, # 6
Step Filtering Width = 23
Lineaments, # 7
Step Filtering Width = 21
Lineaments, # 8
Step Filtering Width = 29
53.5
53.4
53.3
53.2
53.1
-78
-77.8
-77.6
-77.5
-77.4
-77.2
-77

Figure 121

12. Fast Machine Learning with MLP in SFES2D

Here we present an example of machine learning with MLP on satellite and geophysical data sets to improve the resolution of gamma ray targets. The inputs are three RGB color image bands, as well as DEM, Gravity and Magnetic data sets images. The targets are three radionuclide element concentrations, K, eTh and eU. Each of these targets are measured with airborne sensors from different projects, and as a result combining them in unified image produces visible artifacts in the edge of the different projects coordinator. Here we use machine learning with MLP to not only eliminate the previously mentioned artifacts, but also augment the resolution of the gamma ray bands.

This is a super-resolution algorithm based on new advances in machine learning to improve the resolution of several geophysical datasets. The method is based on the automation of Feature extraction for machine learning with the integration of different methods such as continuous wavelet transform (CWT), principal component analysis (PCA), analysis in independent components (ICA) and convolutional neural networks (CNN). Which is a MLP here (or Multilayer Perceptron). It is a hybrid system, named "Spectral Learning", four interconnected modules: a first ICA operator, a CWT, another PCA operator and a MLP (Multilayer Perceptron). To compare the geological targets to the network output, we use an objective function F which we minimize to adjust the PMC weights.

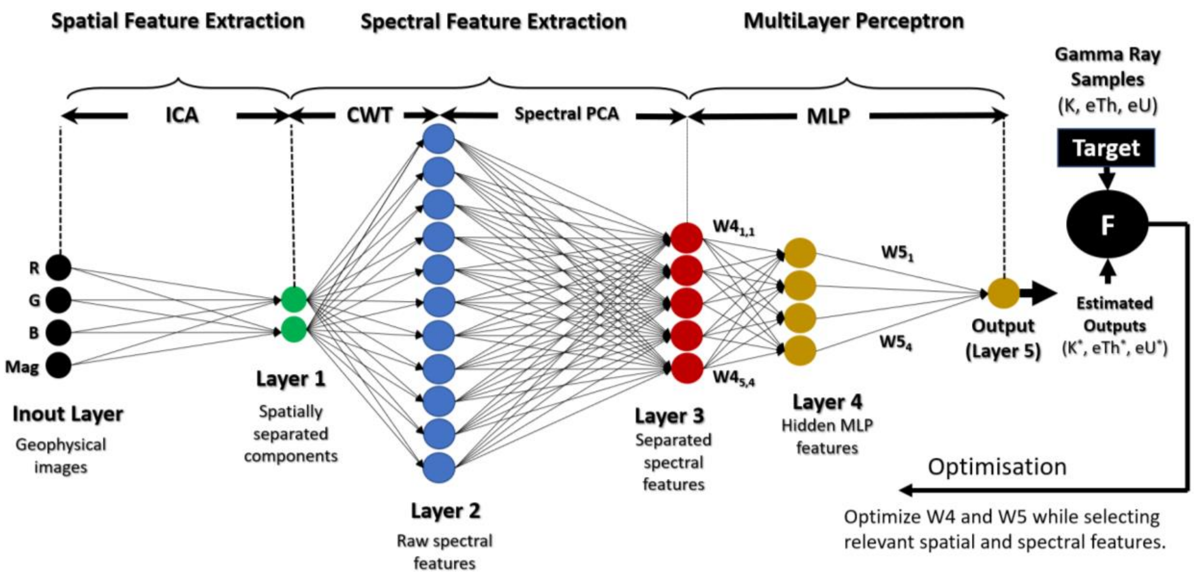


Figure 122

This system is an example of representational learning, as we seek to train the network with all representative features. For less than 5 or 6 features, it's easy to find the best combinations for machine learning. However, with 100 or 1000 features, finding the right combination by testing is difficult. Selecting subsets of Spectral Features allows us to increase the representation or dimensionality as much as possible, then let the computer choose the right Features by emulating natural selection through genetic algorithm optimization.

A standard way to perform a machine learning with MLP in SFES2D is as the following workflow:

- Read the coordinates.

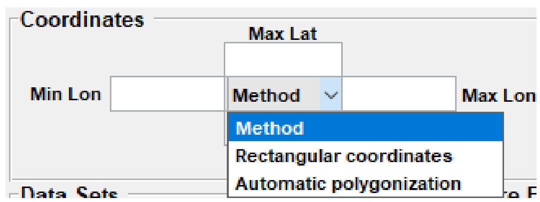


Figure 123

- Set Spacing and Data Filter values. If you do not want to smooth the data sets, just insert zero in the box. Any value greater than 0 will smooth the data sets proportional to the magnitude of the Data Filter.

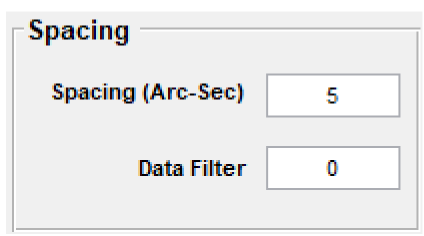


Figure 124

- Push Image/XYZ Data buttons and select the desired data sets.

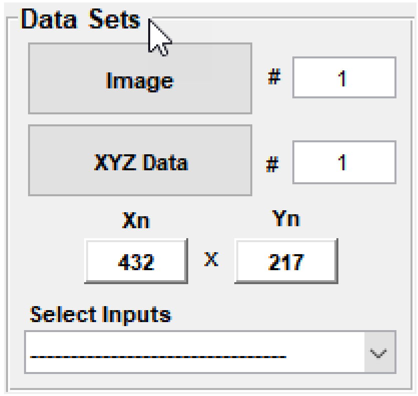


Figure 125

- Users can go on and use only their raw input images for machine learning or continue with Spatial and/or Spectral feature extraction procedures (sections 7 and 8). Either way, the inputs for machine learning can be selected in the box below. For example, here we only ran a fast PCA on the input data sets with an RGB satellite imagery and DEM, Gravity, and Magnetic images, comprising 6 images without dimensionality reduction.
- The study area is in southern parts of the Ontario/Quebec in Canada.

Un exemple

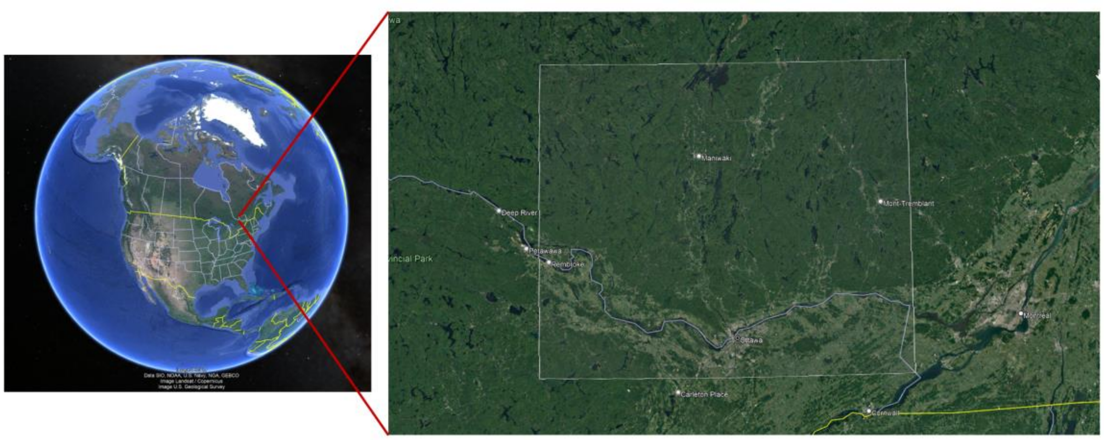
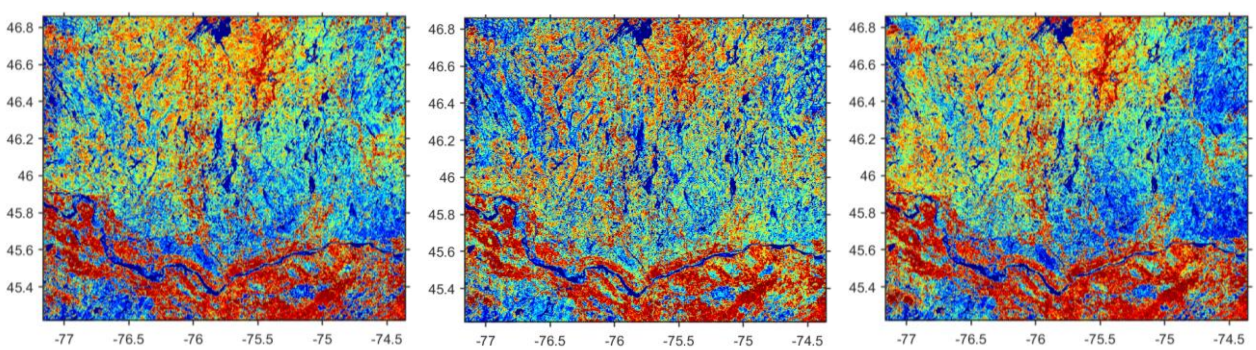


Figure 126

Figure 127

Figure 128

Red Green Blue



DEM Gravity

Magnetic

Figure 129

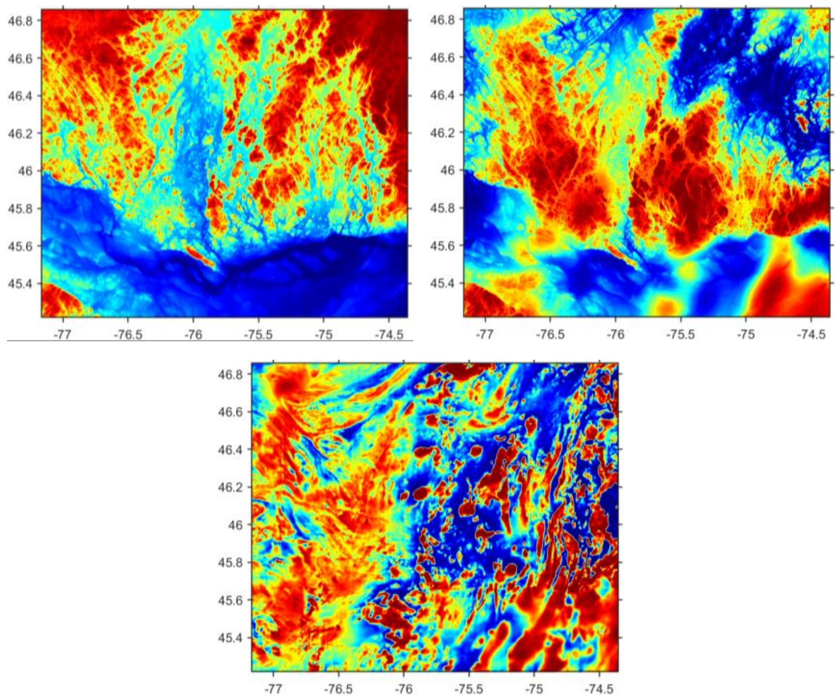


Figure 130

- To upload the target gamma ray data, we use the Validation Data box.

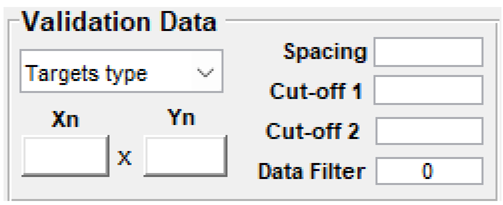


Figure 131

- We need also to define the spacing for target interpolation which is usually smaller or equal to the Data Sets spacing.

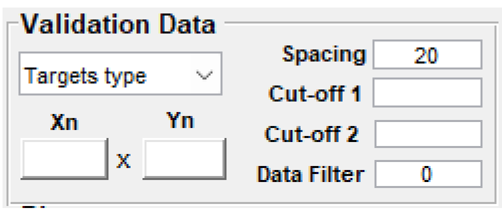


Figure 132

- There are two target types:

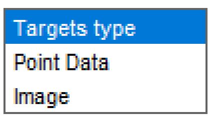


Figure 133

The targets in the form of Point Data are CSV files in XYZ format. The targets in the form of Image also need a coordinate for the georeferencing the target image. A window will open after selecting the Image to upload the coordinates for target georeferencing in a .txt format (for the Data Point form, one need no georeferencing .txt file).

- As soon as selecting Point Data or Image the value inside the Cut-off 1 box is going to be updated. Cut-off 1 determines the buffer zone around the target samples in XY plane.

Figure 134

- The default value of Cut-off 1 factor is unchangeable and is determined in a way it produces the smallest possible buffer around the data points.
- To expand the buffer zone users can increase the Data Filter value. The Cut-off 2 factor also contracts the buffer zone. Manipulating these factors helps to delineate the targets in their real locations and eliminate the places without information about target property in irregularly sampled scenarios. In cases where targets are sampled regularly in a regular grid form, we leave the Cut-off 2 box empty, as in this case. We repeat this process for each radionuclide target and combine them in an RGB image form.

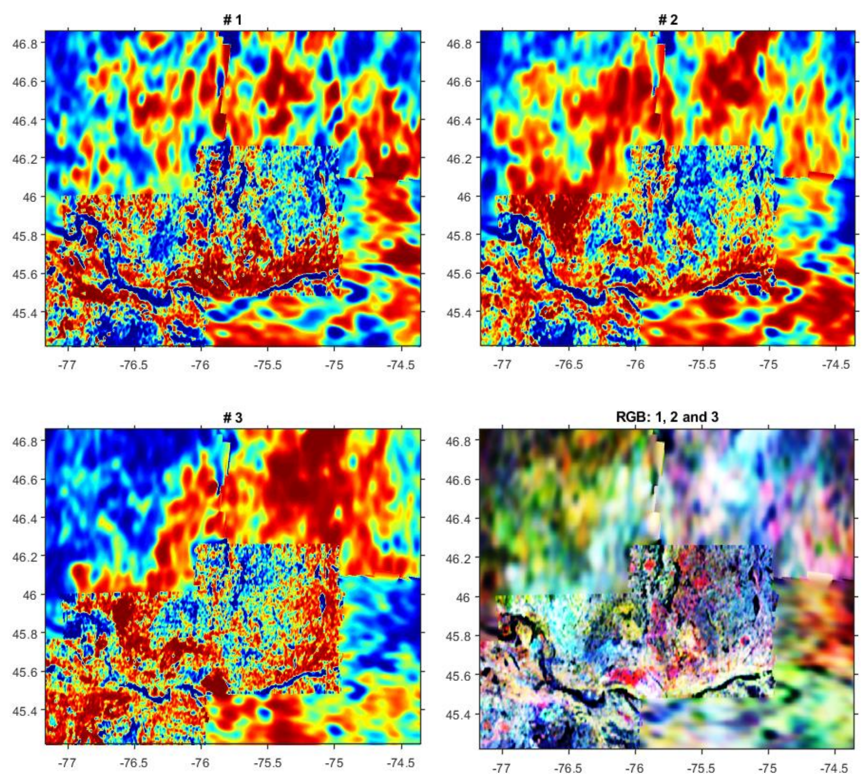


Figure 135

- The source separation procedure starts with an initial ICS with negentropy maximization on RGB image and DEM/Gravity/Magnetic images. Then a CWT with DOG wavelet is ran to increase the dimensionality of the inputs.

Figure 136

- Now we perform a PCA with S-PCA button in SFES2D to reduce dimensionality of the wavelet features to 22.

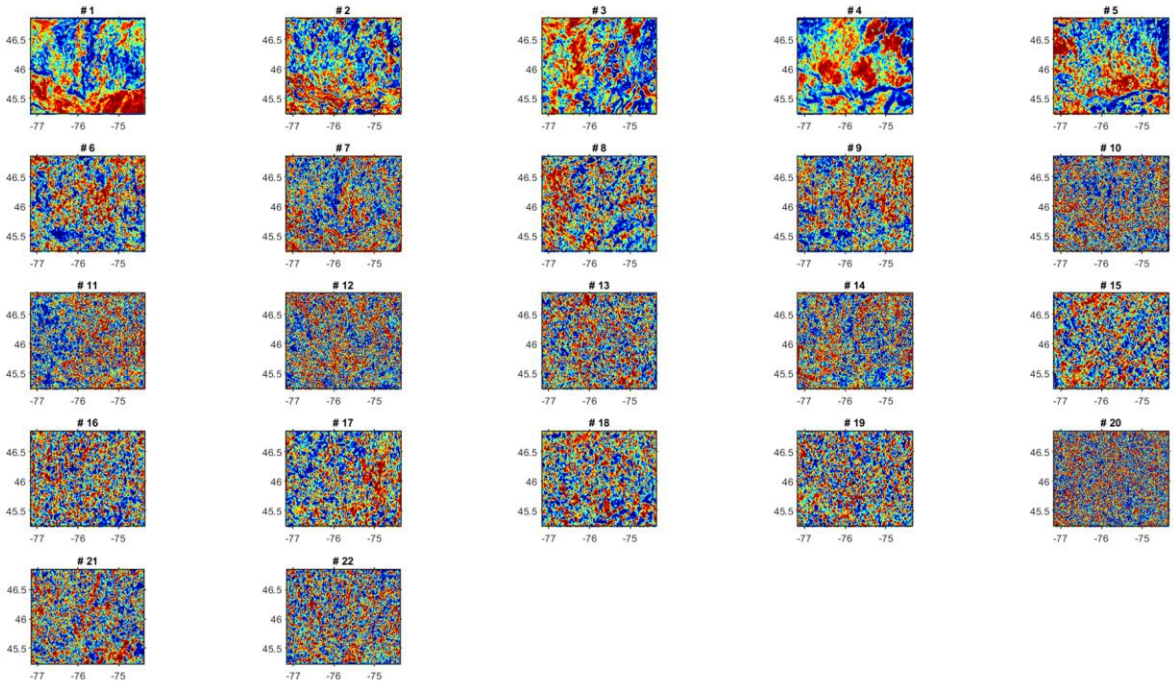


Figure 137

- Then we need to select the Input Features for machine learning:

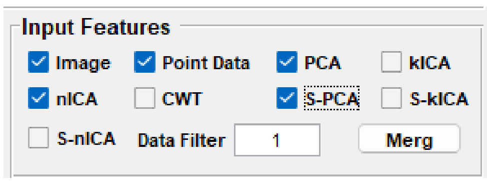


Figure 138

- We set Data Filter to 1 to slightly filter out noises inside the Input Features and smooth them for machine learning training. After that we push the “Merg” button in Input Features box to prepare the data sets for machine learning.
- In the MLP box, the Training Data (%) indicates that 70 percent of target data is selected randomly for training and the rest of 30% is for testing. We used 20 Neurons and Maximum iterations of MLP optimization is set to 40. Now we push MLP.

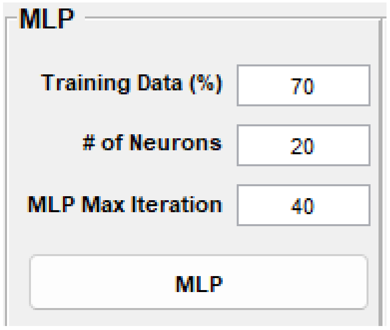


Figure 139

- The algorithm starts to minimize the objective function of MLP weight optimization with the Marquart-Levenberg algorithm.

  Calculation mode: MEX

  Training Custom Neural Network with TRAINLM.

  Epoch 0/40, Time 0.002, Performance 0.44478/1e-12, Gradient 0.85809/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

  Epoch 1/40, Time 0.399, Performance 0.3299/1e-12, Gradient 0.87434/1e-07, Mu 0.0001/10000000000, Validation Checks 0/30

  Epoch 2/40, Time 0.774, Performance 0.27918/1e-12, Gradient 0.60487/1e-07, Mu 0.0001/10000000000, Validation Checks 0/30

  Epoch 3/40, Time 1.159, Performance 0.15597/1e-12, Gradient 0.22402/1e-07, Mu 0.0001/10000000000, Validation Checks 0/30

  Epoch 4/40, Time 1.545, Performance 0.14276/1e-12, Gradient 0.19365/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

  Epoch 5/40, Time 1.946, Performance 0.13007/1e-12, Gradient 0.28222/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

  Epoch 6/40, Time 2.33, Performance 0.11867/1e-12, Gradient 0.15438/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

  Epoch 7/40, Time 2.711, Performance 0.11447/1e-12, Gradient 0.12805/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

  Epoch 8/40, Time 3.12, Performance 0.099864/1e-12, Gradient 0.059212/1e-07, Mu 0.0001/10000000000, Validation Checks 0/30

  Epoch 9/40, Time 3.501, Performance 0.097391/1e-12, Gradient 0.022719/1e-07, Mu 0.0001/10000000000, Validation Checks 0/30

Epoch 10/40, Time 3.876, Performance 0.096515/1e-12, Gradient 0.12037/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

Epoch 11/40, Time 4.285, Performance 0.095296/1e-12, Gradient 0.22278/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 12/40, Time 4.697, Performance 0.090086/1e-12, Gradient 0.093588/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

Epoch 13/40, Time 5.051, Performance 0.088651/1e-12, Gradient 0.10357/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

Epoch 14/40, Time 5.433, Performance 0.088398/1e-12, Gradient 0.17001/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 15/40, Time 5.828, Performance 0.087172/1e-12, Gradient 0.15893/1e-07, Mu 1e-07/10000000000, Validation Checks 0/30

Epoch 16/40, Time 6.301, Performance 0.084623/1e-12, Gradient 0.14037/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

Epoch 17/40, Time 6.709, Performance 0.084008/1e-12, Gradient 0.16884/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 18/40, Time 7.126, Performance 0.082973/1e-12, Gradient 0.12889/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

Epoch 19/40, Time 7.524, Performance 0.082584/1e-12, Gradient 0.1569/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 20/40, Time 7.909, Performance 0.082184/1e-12, Gradient 0.16/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 21/40, Time 8.299, Performance 0.081745/1e-12, Gradient 0.18333/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 22/40, Time 8.699, Performance 0.081248/1e-12, Gradient 0.10098/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 23/40, Time 9.109, Performance 0.080874/1e-12, Gradient 0.1277/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 24/40, Time 9.525, Performance 0.080564/1e-12, Gradient 0.081169/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 25/40, Time 9.901, Performance 0.080258/1e-12, Gradient 0.064325/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 26/40, Time 10.288, Performance 0.080041/1e-12, Gradient 0.048803/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 27/40, Time 10.719, Performance 0.079847/1e-12, Gradient 0.040585/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 28/40, Time 11.126, Performance 0.079694/1e-12, Gradient 0.059663/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 29/40, Time 11.532, Performance 0.079556/1e-12, Gradient 0.055497/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 30/40, Time 11.937, Performance 0.079464/1e-12, Gradient 0.10752/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 31/40, Time 12.347, Performance 0.079271/1e-12, Gradient 0.056546/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 32/40, Time 12.752, Performance 0.079259/1e-12, Gradient 0.15465/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 33/40, Time 13.125, Performance 0.078962/1e-12, Gradient 0.0646/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 34/40, Time 13.543, Performance 0.078638/1e-12, Gradient 0.056807/1e-07, Mu 1e-05/10000000000, Validation Checks 0/30

Epoch 35/40, Time 13.918, Performance 0.078426/1e-12, Gradient 0.025957/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 36/40, Time 14.315, Performance 0.078369/1e-12, Gradient 0.1135/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 37/40, Time 14.726, Performance 0.078127/1e-12, Gradient 0.040808/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 38/40, Time 15.137, Performance 0.07806/1e-12, Gradient 0.10454/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 39/40, Time 15.54, Performance 0.077833/1e-12, Gradient 0.054185/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Epoch 40/40, Time 15.917, Performance 0.077774/1e-12, Gradient 0.099871/1e-07, Mu 1e-06/10000000000, Validation Checks 0/30

Training with TRAINLM completed: Training finished: Reached maximum number of epochs.

- We ran this process for each radionuclide (K, eTh and eU) and the predicted outputs are combined in one single RGB image. Here's a side-by-side comparison below. As can be seen, spectral learning not only increases the resolution, but also easily integrates the different datasets with different flight specifications. The results show that the method is capable of reconstructing three gamma ray images at higher resolutions. The method also helps solve the problem of combining multiple gamma ray data sets with different flight line spacings.

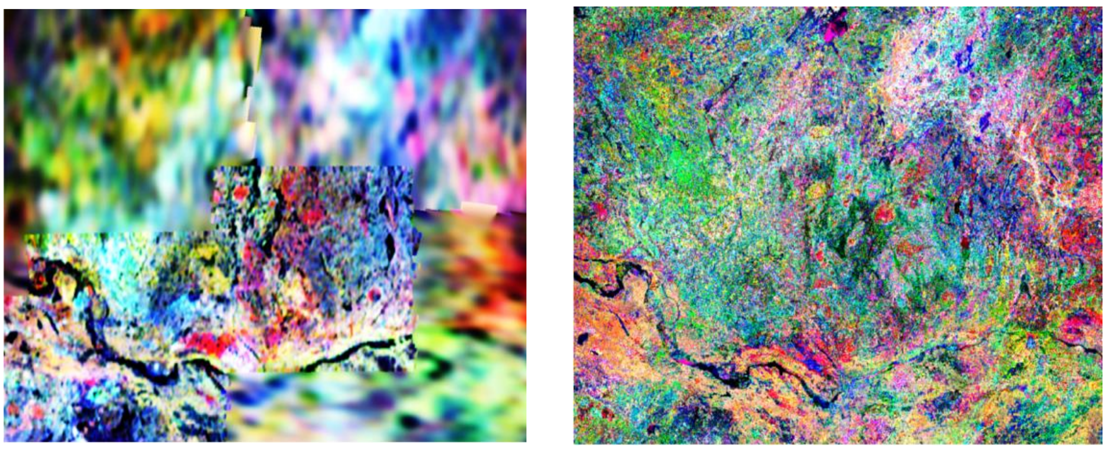


Figure 140

13. Machine Learning by Feature Subset Selection in SFES2D (under progress for ver. 5.2)

This module aims to run predictive modeling on satellite and geophysical images with geological and geoscientific targets.

14. References

Abbassi, B. (2018). Integrated imaging through 3D geophysical inversion, multivariate feature extraction and spectral feature selection (Doctoral dissertation, Université du Québec).

Abbassi, B., 2018, *Integrated imaging through 3D geophysical inversion, multivariate feature extraction, and spectral feature selection*, Ph.D. Thesis, Université du Québec.

Abbassi, B., Cheng, L.-Z., Jébrak, M., & Lemire, D. (2022). 3D geophysical predictive modeling by spectral feature subset selection in mineral exploration. Minerals, 12(10).

Abbassi, B.; Cheng, L.-Z. SFE2D: A Hybrid Tool for Spatial and Spectral Feature Extraction. In Mining Technology; Hammond, A. ,Donnelly, B., Ashwath, N., Eds.; IntechOpen: London, UK, 2021.

Antoine, J.-P., Murenzi, R., Vandergheynst, P., & Twareque Ali, S. (2004). Two-dimensional wavelets and their relatives. Cambridge University Press.

Hyvärinen, A., and E. Oja, 2000, *Independent component analysis: algorithms and applications*, Neural Networks 13, 411–430.

Panagiotakis, C., & Kokinou, E. (2015). Linear pattern detection of geological faults via a topology and shape optimization method. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8(1), 3-11.