APiS (AI Powered intelligence Spectrum analyzer): Hyperspectral Imaging Machine Learning Analyzer App for Mineral Processing

Natsuo Okada, Narihiro Owada, Kaito Takizawa, Brian Bino Sinaice, Okhala Muacanhia, Angesom Gebretsadik, Shunki Tanaka, Kosei Aikawa, Ilhwan Park, Mayumi Ito, Yoko Ohtomo and Youhei Kawamura

Division of Sustainable Resources Engineering, Graduate School of Engineering, Hokkaido University, Kita-13, Nishi-8, Sapporo 060-8628, Japan:
N.O(1): natsuookada01@gmail.com
K.T: takizawa.kaito.b6@elms.hokudai.ac.jp
O.M: okhalaorikamulakha.muacanhia.r1@elms.hokudai.ac.jp
A.G: angstg2007@gmail.com
S.T: tanaka_ramen@eis.hokudai.ac.jp

Graduate School of International Resource Sciences, Department of Earth Resource Engineering and Environmental Science, Akita University, Akita 010-0852, Japan:
N.O(2): owada@gipc.akita-u.ac.jp

Center for Regional Revitalization in Research and Education, Akita University, Akita 010-8502, Japan:
B.B.S: bsinaice@rocketmail.com

Division of Sustainable Resources Engineering, Faculty of Engineering, Hokkaido University, Hokkaido University, Kita-13, Nishi-8, Sapporo 060-8628, Japan (I.P., M.I., Y.O., and Y.K.):
I.P: i-park@eng.hokudai.ac.jp
M.I: itomayu@eng.hokudai.ac.jp
Y.O: ohtomoy@eng.hokudai.ac.jp
Y.K: kawamura@eng.hokudai.ac.jp

Department of Mining Engineering, Aksum University, Aksum, Tigray,7080, Ethiopia:
A.G: angstg2007@gmail.com

National Institute of Advanced Industrial Science and Technology (AIST), Department of Energy and Environment, Environmental Management Research Institute, 16-1, Onogawa, Tsukuba, Ibaraki 305-8569, Japan
K.I: k.aikawa@aist.go.jp

Word count: 5976 words

ABSTRACT:

Hyperspectral data, encompassing hundreds of spectral wavelengths, is a powerful tool for mineral processing, as minerals display unique reflection spectra. Machine learning can identify mineral species from hyperspectral data, but its high dimensionality poses challenges for ordinary users. To address this, we developed APiS (AI Powered intelligence Spectrum Analyzer), an interactive hyperspectral analysis application requiring no coding. APiS offers a user-friendly interface for data visualization, learning data generation, advanced machine learning analysis, and model evaluation. This innovative tool surpasses previous script-based methods, enabling diverse users to conduct sophisticated hyperspectral data analysis and fostering creativity in their work.

Keywords: Spectroscopy, Hyperspectral imaging, Mineralogy, Mineral processing

Highlights:

- ✓ Our proposed APiS (AI Powered intelligence Spectrum analyzer) allows for advanced analysis of hyperspectral data in an intuitive manner.
- ✓ APiS provides powerful support for hyperspectral data visualization, learning data generation, machine learning, and evaluation of learning models.
- ✓ A user-friendly application has been developed with the objective of facilitating the development of hyperspectral sensor-based mineral processing technology.

List of Abbreviations

APiS- AI Powered intelligence Spectrum analyzer

HS- Hyperspectral

RGB- Red, Green, Blue

GUI- Graphical User Interface

CNN- Convolutional Neural Network

NN- Neural Network

SVM- Support Vector Machine

1D- One Dimensional

MAE- Mean Absolute Error

MSE- Mean Squared Error

RMSE- Root Mean Squared Error

1. Introduction

1.1. Background

Hyperspectral data, also known as imaging spectroscopy, is defined as high-dimensional data containing spatial and spectral information [1]. Although it has been extensively studied for its powerful performance in mineral identification when combined with machine learning [2], [3], [4], [5], [6], [7], [8], [9], processing hyperspectral data is challenging due to its enormous nature and the large number of dimensions involved. The identification of minerals is of particular importance in the field of mineral processing. For instance, the use of sensor-based processing equipment can result in cost savings and enhanced efficiency in mineral processing, as it enables the removal of vein stones prior to polishing and flotation.

Consequently, we developed an interactive MATLAB-based analysis script that integrates hyperspectral data and machine learning processing in a highly integrated manner [10]. The script was innovative in that it facilitated the use of hyperspectral data processing by a much broader range of users than was previously possible, as it was previously only accessible to specialists. Nevertheless, there was scope for enhancement in terms of the graphical user interface (GUI) and the ease with which functionality could be extended. However, script-based software required programming knowledge because it was mainly command line operations. In particular, script execution and parameter setting were performed manually, making operations complicated. Furthermore, the software did not support data visualization, which required separate processing on the user side. Additionally, switching between processing functions required the user to execute a new script, and the transition between modules was not clear.

This iteration of the system has been enhanced with further improvements in usability and an expanded range of users who can benefit from it. We have developed a new application named APiS (AI Powered intelligence Spectrum analyzer), which offers significant improvements in usability, efficiency, and functionality. This has been achieved by converting the MATLAB script into an application. Furthermore, new functions have been added and enhanced, including the display of spectra and pseudo-RGB images. The application is designed to be user-friendly and does not require any programming skills. It is tightly integrated with MATLAB, which enables the incorporation of new user-specific analyses.

1.2. APiS Overview

The AI-powered intelligence Spectrum analyzer (APiS) is designed to analyze hyperspectral data with minimal user input, eliminating the need for coding. The user interface is comprised of four modules: (1) visualization, (2) learning, (3) segmentation, and (4) testing. These modules facilitate intuitive operation. As illustrated in Figure 1, the analysis process is structured in accordance with the conventional workflow of machine learning. This functionality enables the user to visualize hyperspectral data, construct training data, select a machine learning model, train the model, and evaluate its performance in accordance with a consistent workflow. This greatly enhances usability and user experience. A particular focus is placed on the integration with machine learning, which enables users to perform classification and regression tasks. Furthermore, 34 machine learning

models, including the most advanced convolutional neural network (CNN) models, are employed for classification, and 27 machine learning models, including CNN, are employed for regression. This facilitates the performance of analysis tailored to the user's needs.

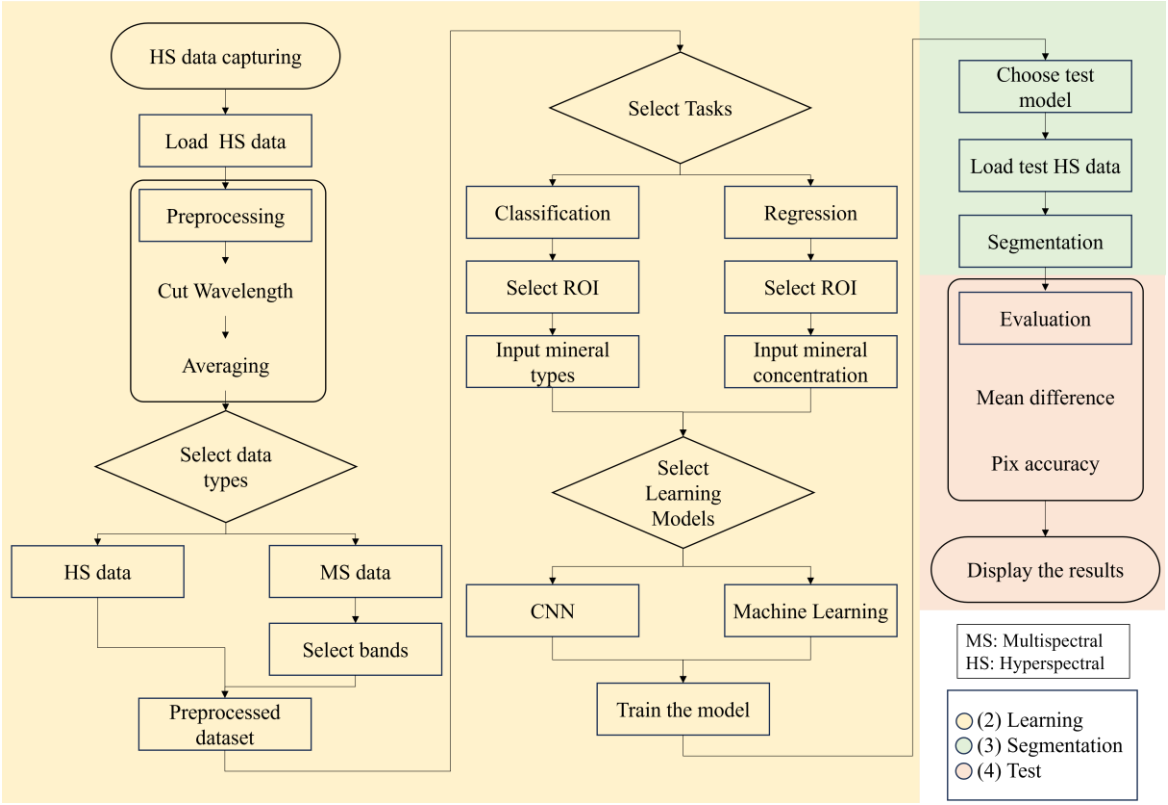

Figure 1 Overview of APiS.

1.3. Related works

Hyperspectral data can be analyzed using commercial software as well as programming languages such as MATLAB and Python. MATLAB provides the Image Processing Toolbox™ Hyperspectral Imaging Library [11], which provides functions and tools for processing and visualizing hyperspectral images [12]. In Python, the Python module Spectral Python (Spy) is provided for processing hyperspectral data, which also has many functions for processing hyperspectral data. As a commercial software, ENVI (Environment for Visualizing Images, Inc., Boulder, CO, USA) is one of the most widely used tools and a leading tool for analyzing hyperspectral data.

Nevertheless, a sophisticated comprehension of programming languages is essential for the effective utilization of the tools provided in MATLAB and Python. Even commercial software, such as ENVI, is labor-intensive when used for processing large data sets [13]. A number of tools for the analysis of hyperspectral data have been proposed, including HSI-PP and HYPER-Tools [14], [15]. While these tools offer advanced and user-friendly hyperspectral data processing capabilities, they have not yet been integrated with deep learning models, which have been growing rapidly in recent years.

This study proposes a user-friendly hyperspectral data analysis application that does not require the use of advanced programming techniques. The application's design incorporates the integration of hyperspectral data and machine learning, thus enabling not only data visualization but also the construction of training data, training, and the evaluation of training models. The application facilitates the accessibility of hyperspectral data analysis technology, which was previously confined to a select group of experts with programming expertise.

2. Methodology

2.1. Development Environments

The primary development environment for this project was MATLAB (R2023a). The Computer Vision Toolbox was employed for this project. The toolboxes used were Computer Vision Toolbox: Version 10.4 (R2023a), Deep Learning Toolbox: Version 14.6 (R2023a), Image Processing Toolbox: Version 11.7 (R2023a), Parallel Computing Toolbox: Version 7.8 (R2023a), Signal Processing Toolbox: Version 9.2 (R2023a), Statistics and Machine Learning Toolbox Version 12.5 (R2023a) were used. The MATLAB App Designer, as illustrated in the accompanying Figure 2, was employed to construct the application. The Microsoft Windows 11 Home Version 10.0 operating system was utilized.

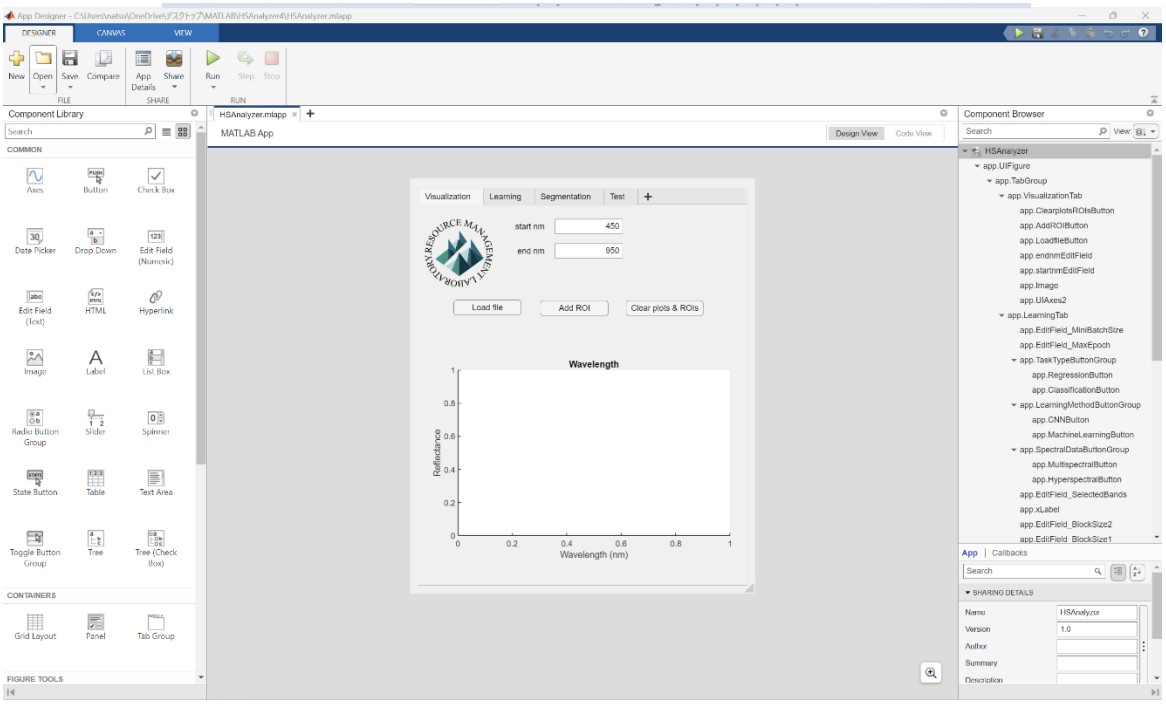

Figure 2 Overview of MATLAB App Designer.

2.2. UI / UX Design

The APiS system places a strong emphasis on straightforward and user-friendly operation, with all user interface components conveniently arranged along the analysis flow. Each system module can be accessed by changing tabs and showed in Figure 3.

(1) Visualization: This module displays the average spectra of hyperspectral data. The user is afforded the opportunity to select the region of interest (ROI) and the desired wavelength width for display. Overview of this tab is displayed in Figure 3(a).
(2) Learning: In this module, the user is afforded the opportunity to select the hyperspectral data to be used as training data by ROI, to select the wavelength region to be used, to select the block size of the averaging filter, to select the analysis of hyperspectral or multispectral data, to select the machine learning method, and to select the classification or regression task. In the case of multispectral data, the user is permitted to enter any number of bands and perform the analysis using only the specified bands. When the convolutional neural network (CNN) is selected for machine learning, the user may enter the number of epochs and mini-batches. Overview of this tab is displayed in Figure 3(b).
(3) Segmentation: In the preceding module, the training data settings and training parameter values are input. In this module, machine learning is performed using those values. The machine learning process is tightly integrated with the MATLAB Classification Learner app or the Regression Learner app [16], [17]. Once the machine learning process is complete, segmentation of the test hyperspectral data is performed. Overview of this tab is displayed in Figure 3(c).
(4) Test: In this module, the test images segmented in module (3) are evaluated using ground truth images, and the mean difference and pixel accuracy are calculated. Overview of this tab is displayed in Figure 3(d).

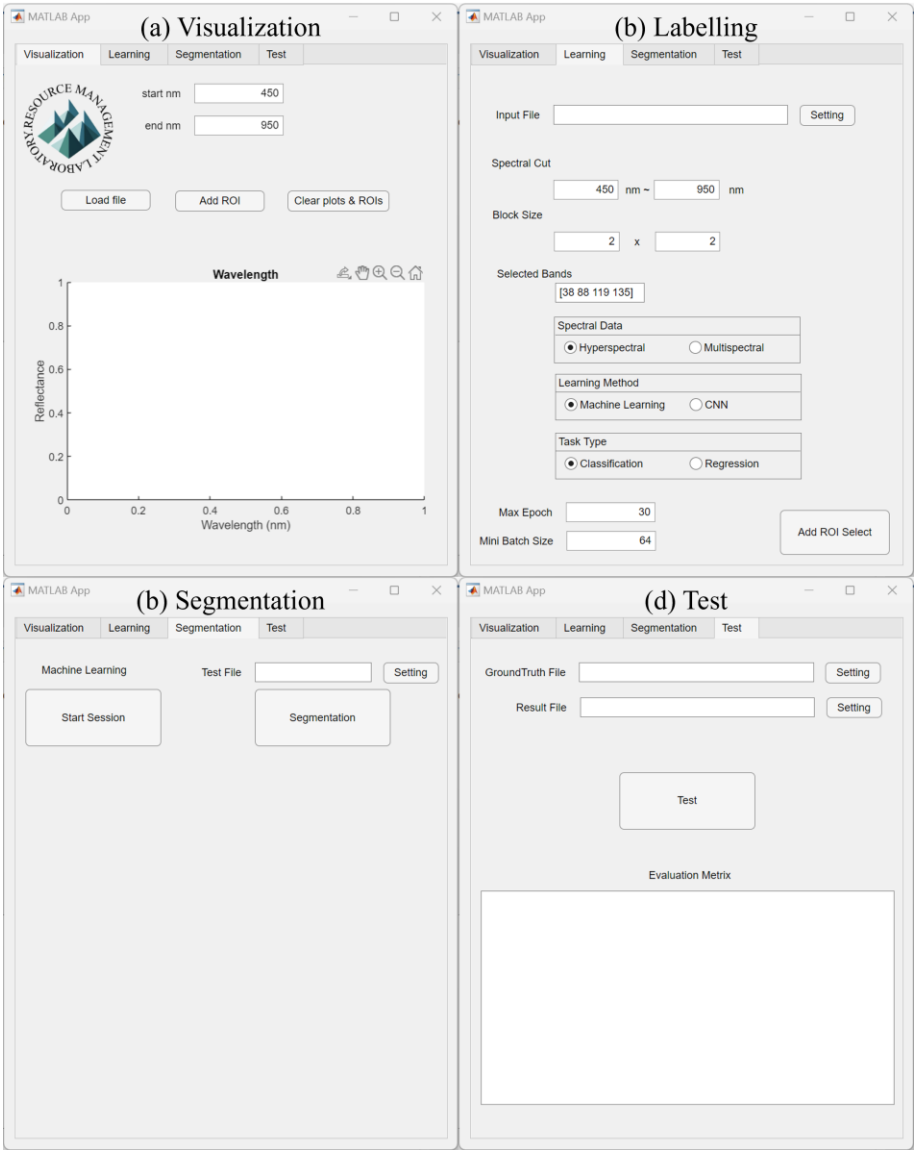

Figure 3 Overview of APiS app.

2.3. Implementation of Algorithm

Hyperspectral data analysis is conducted on one-dimensional hyperspectral wavelength data. By selecting regions of interest (ROIs), any location in the hyperspectral data is extracted, averaged, and stored as data for machine learning with label names, as illustrated in Figure 4.

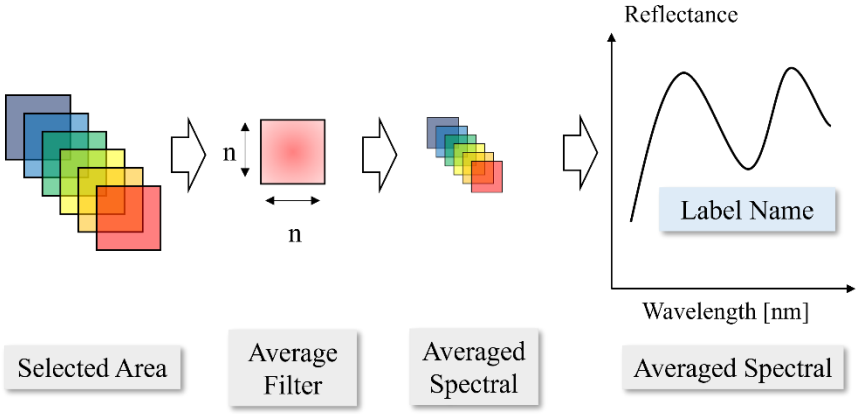

Figure 4 ROI selection and averaging spectral process.

The stored training data are subjected to analysis using convolutional neural networks (CNNs) or one of the numerous machine learning methods available in the MATLAB Classification Learner App, Regression Learner App, and others. The user may select any machine learning model or utilize and contrast all learning models for training. The CNN employs a 1D-CNN model developed by the authors, and a list of machine learning models for classification and regression is presented in Table 1 and Table 2.

Table 1 List of machine learning algorithms for classification task.

Name	Name	Name	Name	Name
CNN	Efficient Linear SVM	Medium Gaussian SVM	Weighted KNN	Medium Neural Network
Fine Tree	Gaussian Naïve Bayes	Coarse Gaussian SVM	Boosted Trees	Wide Neural Network
Medium Tree	Kernel Naïve Bayes	Fine KNN	Bagged Trees	Bilayered Neural Network
Coarse Tree	Linear SVM	Medium KNN	Subspace Discriminant	Trilayered Neural Network
Linear Discriminant	Quadratic SVM	Coarse KNN	Subspace KNN	SVM Kernel
Quadratic Discriminant	Cubic SVM	Cosine KNN	RUSBoosted Trees	Logistic Regression Kernel
Binary GLM Logistic Regression	Fine Gaussian SVM	Cubic KNN	Narrow Neural Network	

Table 2 List of machine learning algorithms for regression task.

Name	Name	Name	Name	Name

CNN	Medium Tree	Medium Gaussian SVM	Exponential GPR	Trilayered Neural Network
Linear Regression	Coarse Tree	Coarse Gaussian SVM	Rational Quadratic GPR	SVM Kernel
Interactions Linear	Linear SVM	Boosted Trees	Narrow Neural Network	Least Squares Regression Kernel
Robust Linear	Quadratic SVM	Bagged Trees	Medium Neural Network	
Stepwise Linear	Cubic SVM	Squared Exponential GPR	Wide Neural Network	
Fine Tree	Fine Gaussian SVM	Matern 5/2 GPR	Bilayered Neural Network	

2.3.1. Classification vs Regression

Classification and regression are two common methods of machine learning. For example, classification tasks include mineral species identification, while regression tasks include mineral composition concentration estimation. Various machine learning models have been proposed for each of these tasks. Mathematically, classification is the mapping of a function (f) as target, label, or category from input variables (X) to output variables (Y). This process can be applied to both structured and unstructured data in order to predict the class of provided data points [18]. A significant challenge in machine learning and statistics is regression analysis. The overarching objective of regression research is to infer a real-valued function, or regression function, whose values represent the average of a response or output variable that is dependent on one or more independent variables, also known as input variables [19]. In general, classification is employed to predict a label, whereas regression is typically utilized to predict a quantity [20].

2.3.2. 1D-CNN

A 1D CNN was viable for this tool since the spectral data handled by this tool is 1D data, making a low-cost implementation possible [10]. Let the input signal x be of length n and the convolution kernel *w* be of length *k*. Furthermore, if the stride is *s* and the bias is *b*, the *i*-th element of the output *y* is computed as

$$y_i = \sum_{j=0}^{k-1} w_j x_{i \cdot s+j} + b$$

Batch normalization and *ReLU* activation functions are applied to the output of convolution layers. Letting *z* be the output of *ReLu*, the formula is shown below.

$$z_i = ReLU(y_i) = \max(0, y_i)$$

Then, in the pooling layer, the Max pooling process is performed, and p is output.

$$p_i = \max_{j=0,\ldots,k-1} z_{i\cdot s+j}$$

The model then employs a 1-dimensional global average pooling layer, leading to a final output layer consisting of two fully connected layers. The output of fully connected layers is shown below. Here W is weight of fully connected layer and b is bias.

$$\sigma_j = activation(\sum_i W_{ji} p_i + b_j)$$

The output layer applies a softmax activation function, and the model performs classification tasks using cross-entropy loss.

$$\widehat{y_j} = \frac{\exp(\sigma_j)}{\sum_k \exp(\sigma_k)}$$

The regression task adopted the same architecture except for the last layer.

2.3.3. Decision Tree

The use of decision tree (DT) techniques for the construction of classification models has gained considerable traction in recent times. This is due to the fact that these models are readily comprehensible and exhibit a high degree of fidelity to the processes of human thought. [21].

The DT categorizes the instances by sorting the tree from the root node to a few leaf nodes. The categorization process begins at the tree's root node and progresses along the branch corresponding to the attribute value. This involves examining the attribute that the node defines. The most commonly employed criteria for splitting are "gini" for the Gini impurity and "entropy" for the information gain, which can be expressed mathematically as follows: [18].

$$Engropy: H(x) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i)$$

$$Gini(E) = 1 - \sum_{i=1}^{c} p_i^2$$

2.3.4. Linear discriminant

Discriminant analysis takes a generalized form of Fisher's linear discriminant [22]. While linear discriminants can be used to distinguish between a multitude of classes of patterns, they are perhaps most frequently employed in situations where there are just two classes. The standard for linear discriminating in multidimensional space between two classes is the Fisher discriminant. This is because it is based solely on the first and second moments of each distribution, which allows it to be calculated very quickly. Furthermore, it has been demonstrated to optimize a separation measure that is independent of the distribution type [23].

Here, considering 2-class classification, the total within-class covariance matrix S_w is expressed as follows, where w is the weight vector [22].

$$w \propto S_W^{-1}(m_2 - m_1)$$

where the mean vector of class 1 is m_1 and the mean vector of class 2 is m_2.

2.3.5. Logistic Regression

Statistical models known as logistic regressions are employed to delineate the relationship between an independent variable and a qualitative dependent variable. When categorical outcomes are studied using logistic regression models, the model is referred to as a binary logistic model, as the outcome is typically binary [24].

Here, the dataset $\{ \phi_n, t_n \}$ ($t_n \in \{0, 1\}$ and $\phi_n = \phi(x_n)$, with $n = 1, \ldots N$), the likelihood function $p(t|w)$ is expressed as follows [22].

$$p(t|w) = \prod_{n=1}^{N} y_n^{t_n}\{1 - y_n\}^{1-t_n}$$

where $t = (t_1, \ldots, t_N)$ and $y_n = p(C_1|\varphi_n)$.

2.3.6. Naïve Bayes

The Naive Bayes Classifier is a probabilistic classification mechanism that derives from Thomas Bayes' posthumous theory, the Bayesian Theorem. Some mathematical operations are carried out to convert joint probabilities into the multiplications of prior probabilities and conditional probabilities in order to make this mapping probabilistically calculable [25].

Here, with respect to Bayesian probabilities, the $P(w|D)$ posterior probability is shown as follows [22].

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)}$$

where $P(w)$ be the prior probability distribution, Observed data $D = \{t_1, \ldots, t_N\}$, and Conditional probability $p(D|w)$.

2.3.7. Support Vector Machine (SVM)

Support Vector Machines (SVMs) represent a classical machine learning technique that continues to offer potential solutions to the challenges posed by big data classification problems. In particular, it can facilitate the development of multidomain applications in a big data environment [26].

Here, with respect to kernel funcation K, it is expressed using the nonlinear vector function φ as follows [27].

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$$

where x is an n-dimensional number space. Furthermore, the SVM decision function is shown using the Lagrange multiplier λ_i as follows.

$$f(x) = sign\left(\sum_i \lambda_i y_i K(x_i, x_j) + b\right)$$

2.3.8. Nearest Neighbor

The nearest neighbor (NN) technique is a relatively straightforward approach that has proven to be highly efficient and effective in a number of fields, including pattern recognition, text categorization, and object recognition [28].

The nearest neighbor search in a metric space can be defined as follows [29]. Given a set of points $P=\{p_1,p_2, \cdots,p_n\}$ in a metric space M and a query point $q \in M$, find the element $NN(q,P) \in P$ that is the closest to q with respect to a metric distance $d:M \times M \rightarrow R$:

$$NN(q, P) = argmin_{x \in R} d(q, x)$$

2.3.9. Ensemble

An ensemble is a group of individual classifiers that collectively make predictions, which are then aggregated to make predictions for new instances. It is sometimes referred to as a committee or multiple classifiers [30]. By combining the predictions of several models, these strategies train multiple models and increase the accuracy of a single model [31].

2.3.10. Neural Network

Computer programs or algorithms known as artificial neural networks are inspired by efforts to simulate the behavior of nerve cells. They have been employed in a number of different fields, including the physical and chemical sciences, for the purposes of pattern recognition, classification, and optimization [32].

Here, given *{Wi}* is coefficient, *J=1,…,M* and super print *(1)* indicates that the corresponding parameters are in the first 'layer' of the network. We shall refer to the parameters $w(1)_{ji}$ as weights and the parameters $w(1)_{j0}$ as biases[22].

$$y_k(x,w) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right)$$

Here, y_k is network outputs, and w is the set of all weight and bias parameters have been grouped together into a vector.

3. Case study and experimental procedures

3.1. Material settings

In this study, a total of four mineral samples, namely sphalerite, pyrite, chalcopyrite, and galena, were prepared and utilized as experimental samples for the classification and regression tasks. The samples were pulverized and agitated to ensure uniform distribution of the components.

The use of APiS enables the reading of hyperspectral data and the display of the spectrum of a portion of any wavelength band, as illustrated in Figure 5. Upon the user selecting an ROI on the displayed pseudo-RGB image, the average wavelength spectrum of that portion is displayed on APiS. Figure 6 presents a magnified view of the spectrum displayed on APiS. The horizontal axis represents the wavelength, and the vertical axis represents the reflection intensity. This figure illustrates that pyrite and chalcopyrite exhibit similar spectral shapes, whereas the other mineral samples display distinct spectra.

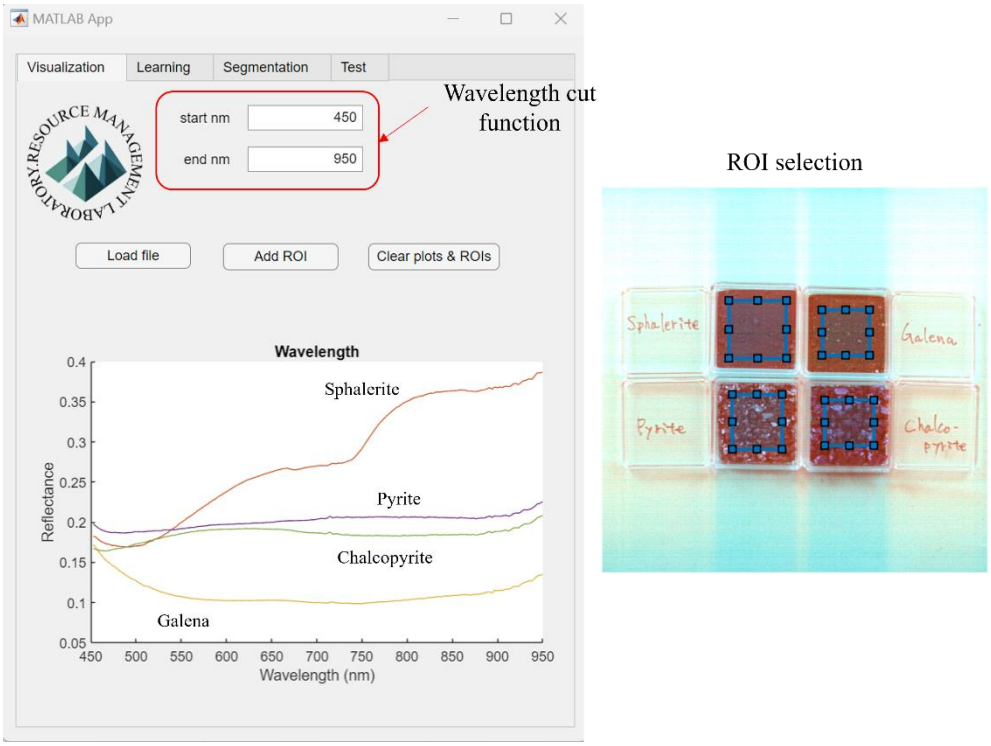

Figure 5 How to show spectral using the App.

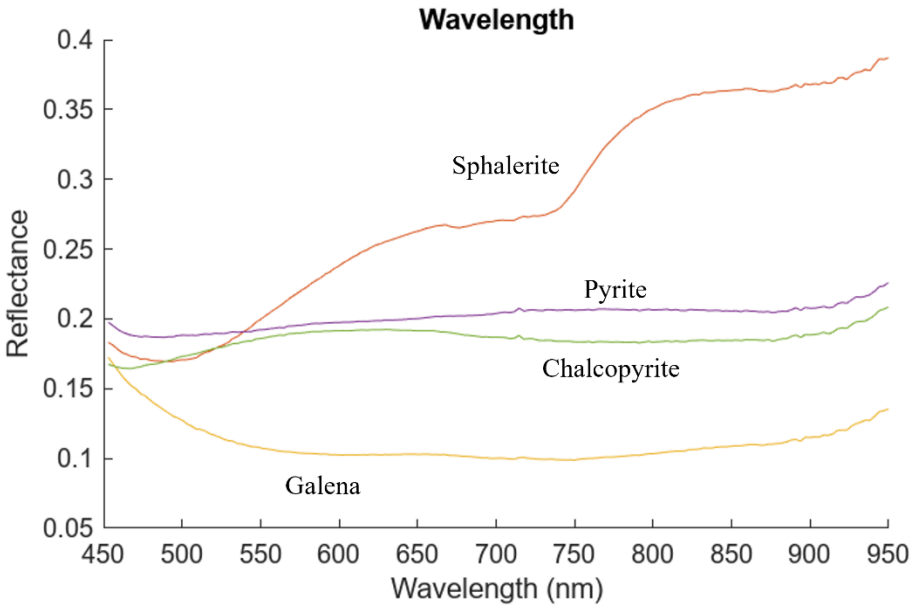

Figure 6 Spectral features of the samples.

3.2. Dataset preparation

The classification task was carried out on the four mineral samples produced. The construction of the labelled datasets necessary for machine learning can be accomplished via the Learning module of this application. As illustrated in Figure 7, the construction of the training data can be completed by entering an arbitrary label name in the dialog box and selecting the ROI of the corresponding region. In this experiment, six classes (background,

plastic case, pyrite, sphalerite, galena, and chalcopyrite) were designated as label names, and a total of 258,890 hyperspectral training data sets were obtained. Subsequently, cross-validation was employed to determine the proportion of teacher data, data for validation, and test data, which was set at 80%, 10%, and 10%, respectively.

Figure 7 (a) input label name and (b) select label area.

4. Results and discussions

4.1. Classification task

4.1.1. Machine learning (except CNN)

Figure 8 presents the outcomes of training 33 machine learning models, excluding those based on convolutional neural networks (CNNs), on data created using APiS. The blue bars indicate the accuracy on the validation data, and the orange bars indicate the accuracy on the test data. The results demonstrate that 28 of the training models were able to classify the test data with more than 90% accuracy. Figure 9 depicts the confusion matrix for Quadratic Discriminant, which exhibited particularly robust performance. The horizontal cells indicate the predicted class, while the vertical cells indicate the true class.

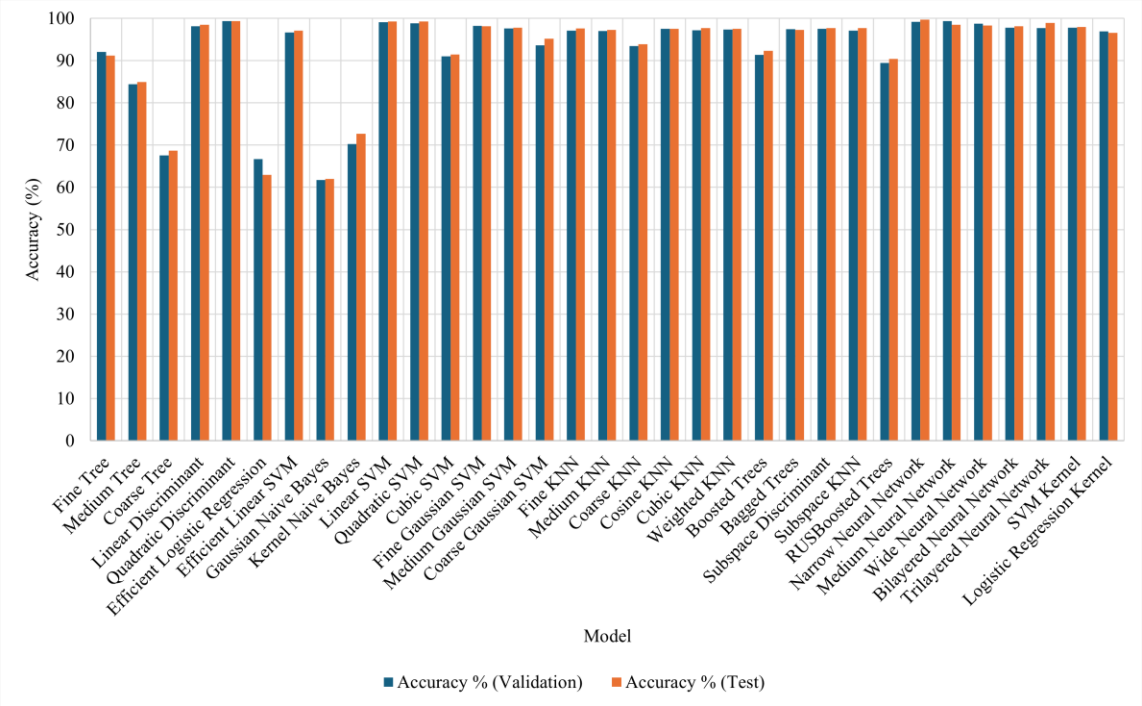

Figure 8 Result of machine learning for classification task.

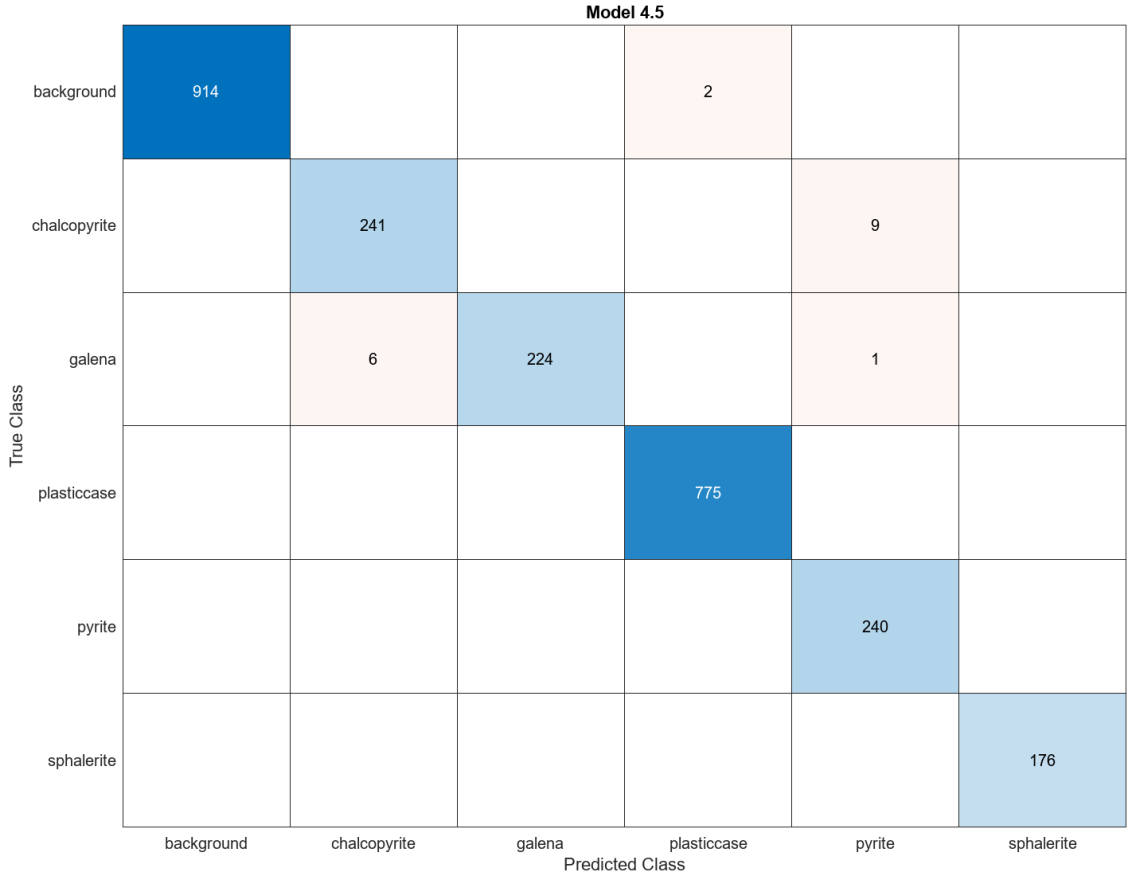

Figure 9 Confusion matrix of Quadratic Discriminant analysis.

Figure 10 presents the outcomes of segmentation on test hyperspectral data utilizing the Quadratic Discriminant Model. The left-hand side of the figure (a) depicts the RGB image of the test data, while the right-hand side (b) shows the result of segmentation on the RGB image. The classification of mineral classes other than chalcopyrite also tends to be generally accurate. Conversely, only chalcopyrite exhibits a low degree of discrimination performance. This is believed to be due to the similarity of the spectra of chalcopyrite and pyrite, as illustrated in Figure 6.

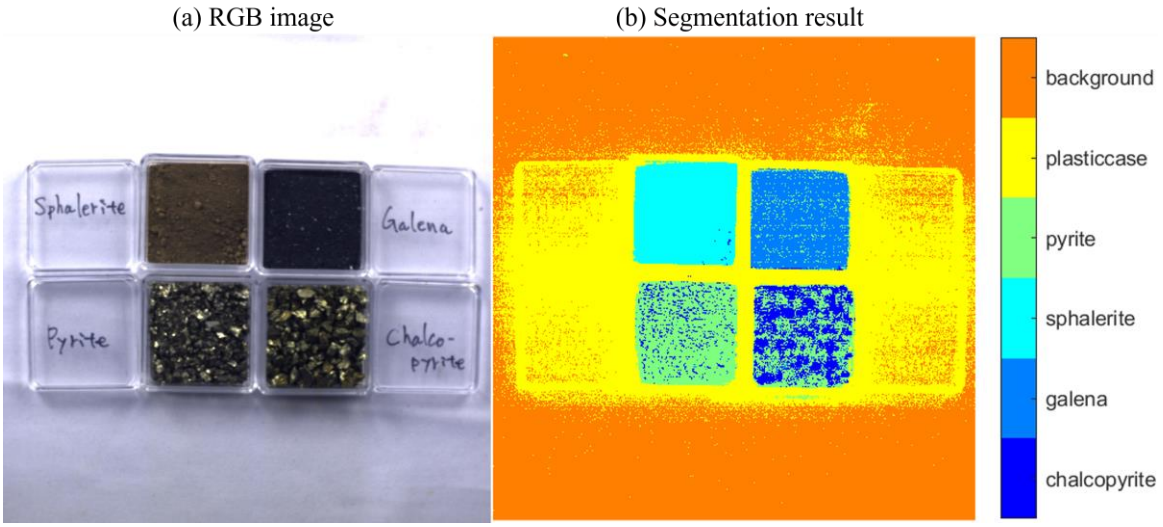

Figure 10 Segmentation result.

Figure 11 presents the quantitative evaluation of the segmentation results obtained from the Test tab. Upon loading and executing the created ground truth and segmentation results, a difference image between them is displayed. The magenta and green areas indicate a significant discrepancy in color between the segmentation results and the ground truth, as well as a substantial misclassification in the area corresponding to chalcopyrite.

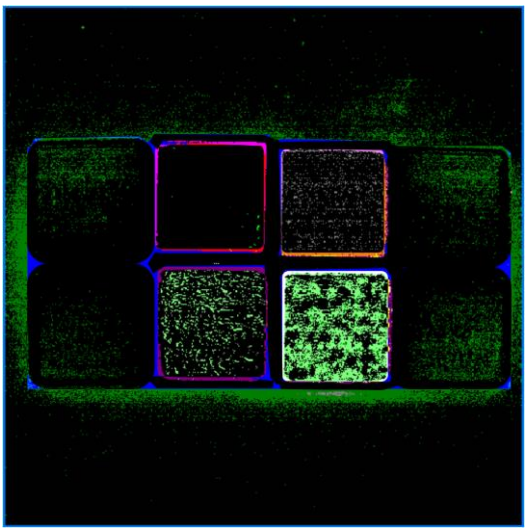

Figure 11 Difference between Ground Truth and segmented results.

4.1.2. CNN

The same dataset utilized in Section 4.1.1 was employed in this analysis, which was conducted using a one-dimensional convolutional neural network (1DCNN) to examine one-dimensional wavelength data. Figure 12 illustrates the training progress. The upper blue plot depicts the accuracy against the teacher data, while the black plot illustrates the accuracy against the validation data. The lower orange plot depicts the training loss. The training results indicate that the training process converged with an accuracy of 91.89%. Figure 13 presents the confusion matrix.

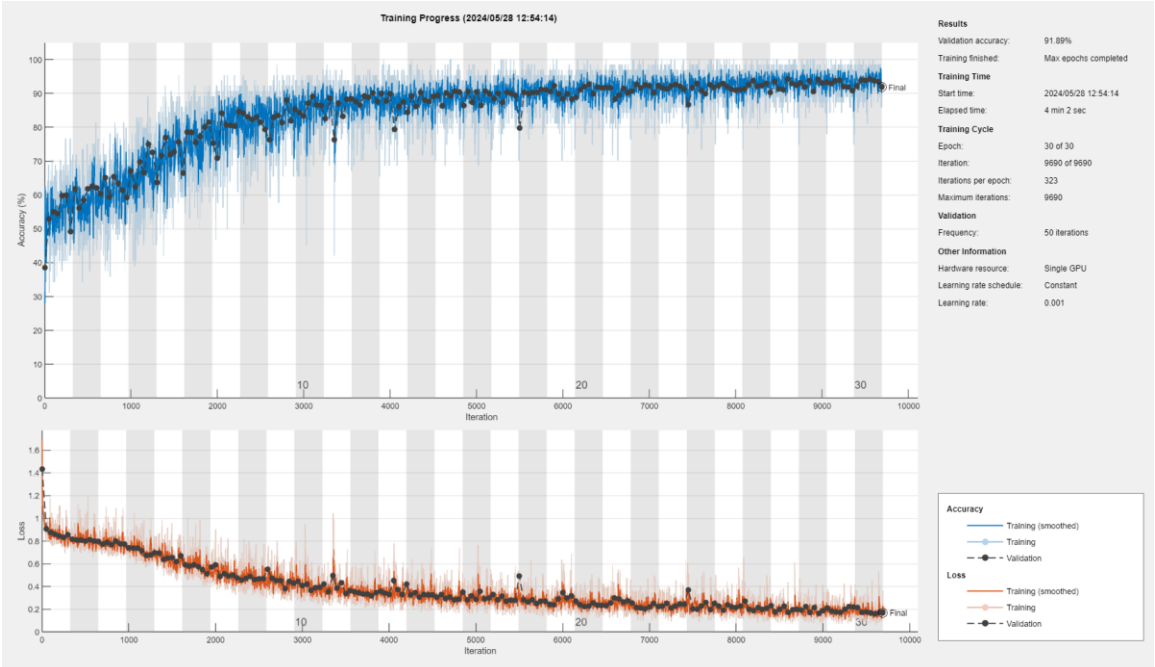

Figure 12 Training progress of CNN.

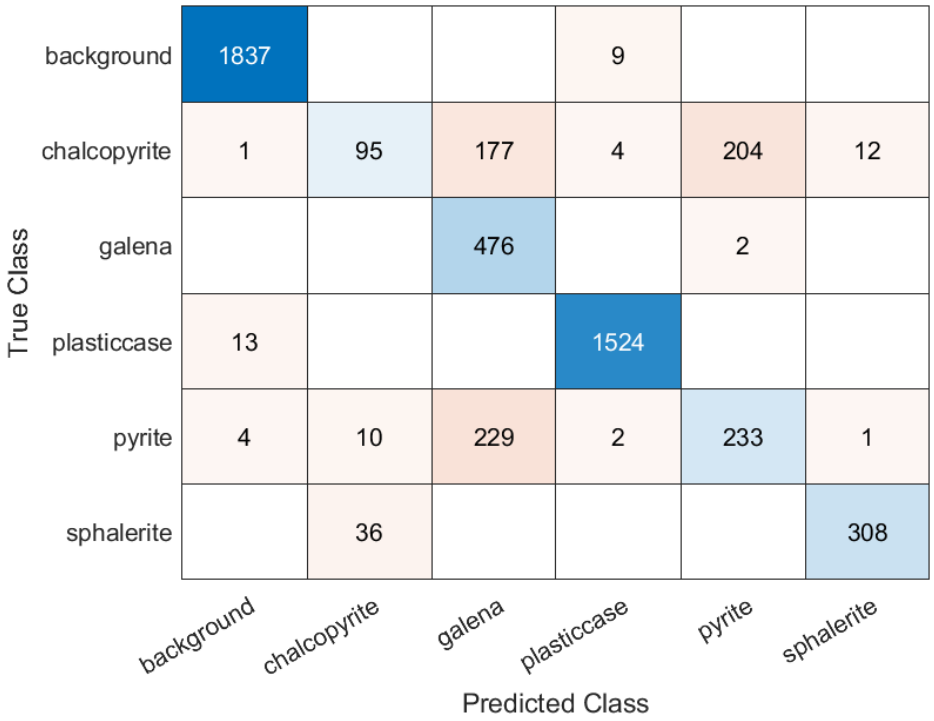

Figure 13 Confusion matrix of CNN.

The test hyperspectral data was subjected to segmentation using a trained convolutional neural network (CNN). Figure 14 illustrates the outcomes, which demonstrate that chalcopyrite and pyrite are incorrectly identified, a phenomenon that was also observed when the quadratic discriminant was employed. The remaining classes are classified with high accuracy. The observed misclassification is not indicative of a deficiency in APiS performance, but rather, it is a consequence of the limitations inherent to machine learning. To enhance the classification efficacy, the quantity of the training data set can be augmented or the parameter values of machine learning can be modified.

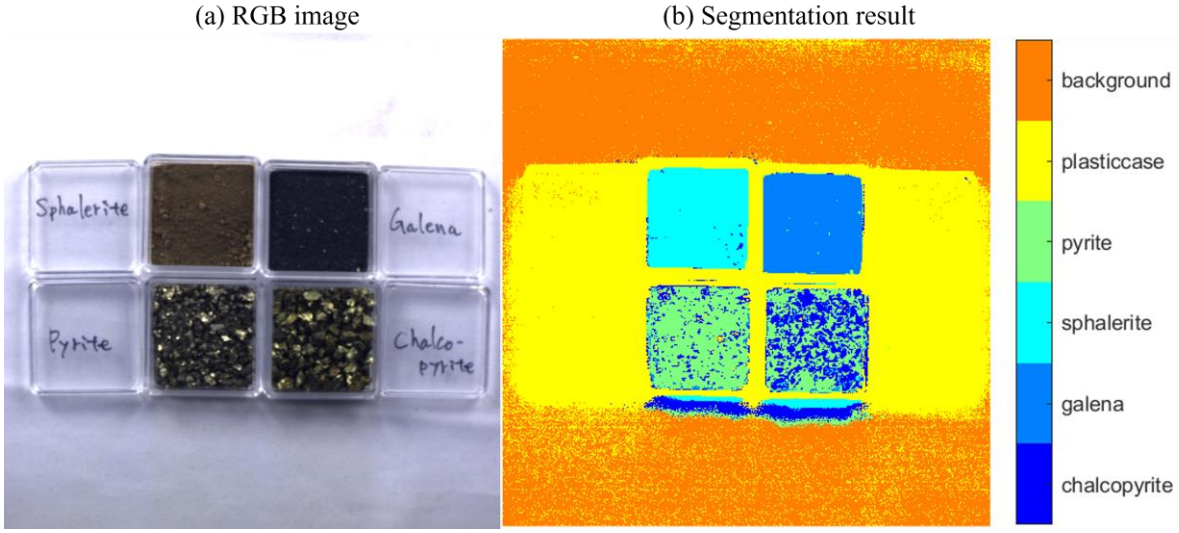

Figure 14 Segmentation result of CNN.

It is possible to evaluate the segmentation results from the Test module. Figure 15 depicts the difference image between the segmentation result image and the ground truth. This image depicts the discrepancy between the user-generated ground truth and the segmented image, with green or magenta indicating a significant divergence. Conversely, the absence of color indicates that there is no difference between the Ground Truth and the segmented image.

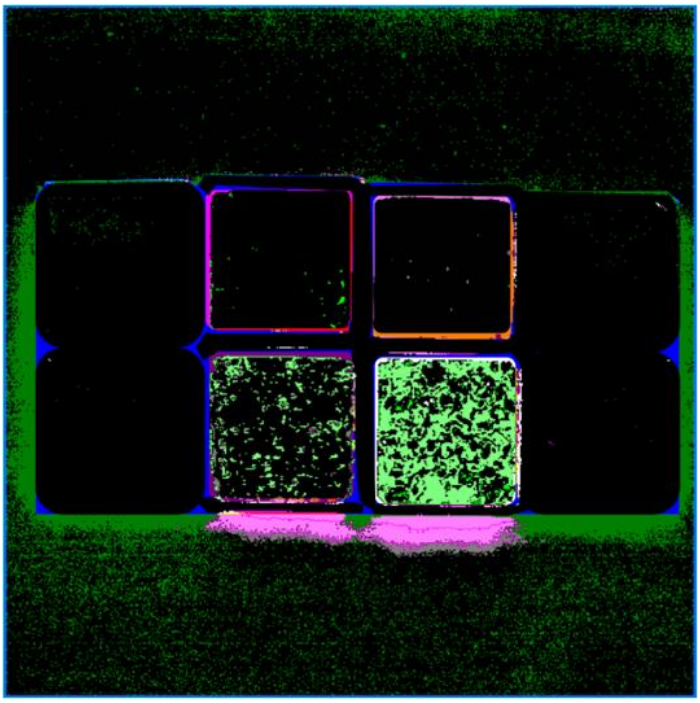

Figure 15 Difference between ground truth and segmented result.

4.2. Regression Tasks

Samples of powdered galena and quartz were prepared as samples for regression analysis, with the galena and quartz being mixed by mass percent. As illustrated in Figure 16, samples were prepared in 10% increments by mass percentage, and each sample was photographed using a hyperspectral camera. Subsequently, as shown in Figure 17, the region of interest (ROI) was selected and labelled on the hyperspectral images using the Learning Module function of APiS. The label name was designated as the mass percentage of galena.

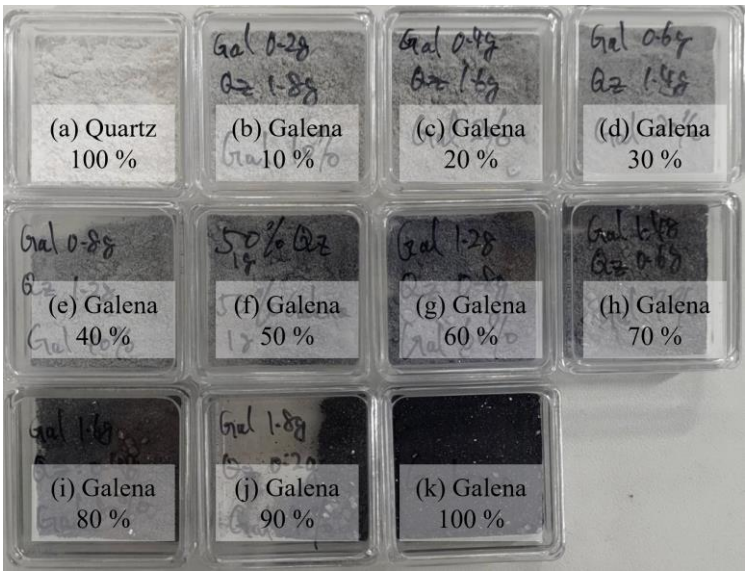

Figure 16 Samples for regression analysis.

Figure 17 Labelling and ROI selection.

4.2.1. Machine learning (except CNN)

Once the training dataset had been constructed, the 25,889 spectral data points corresponding to the portion selected by ROI in machine learning, excluding those processed by CNN, were subjected to analysis. A cross-

validation approach was employed, with the proportions of training, validation, and test data set at 80%, 10%, and 10%, respectively. Figure 18, Figure 19, Figure 20and Figure 21 presents the values of Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R Squared for each learning model. Each mathematical model is expressed as follows. Here, X_i is the predicted i^{th} value and the Y_i element is the actual i^{th} value [33].

Here, MAE is represented as follow.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|X_i - Y_i|$$

MSE is represented as follow.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2$$

RMSE is represented as follow.

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

And R squared is represented as follow.

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(X_i - Y_i)^2}{\sum_{i=1}^{m}(\bar{Y} - Y_i)^2}$$

Here, for the Fine Gaussian SVM, which showed the highest performance in RMSE, a figure of predicted responses vs. true responses was created in Figure 22 to evaluate the performance of the model.

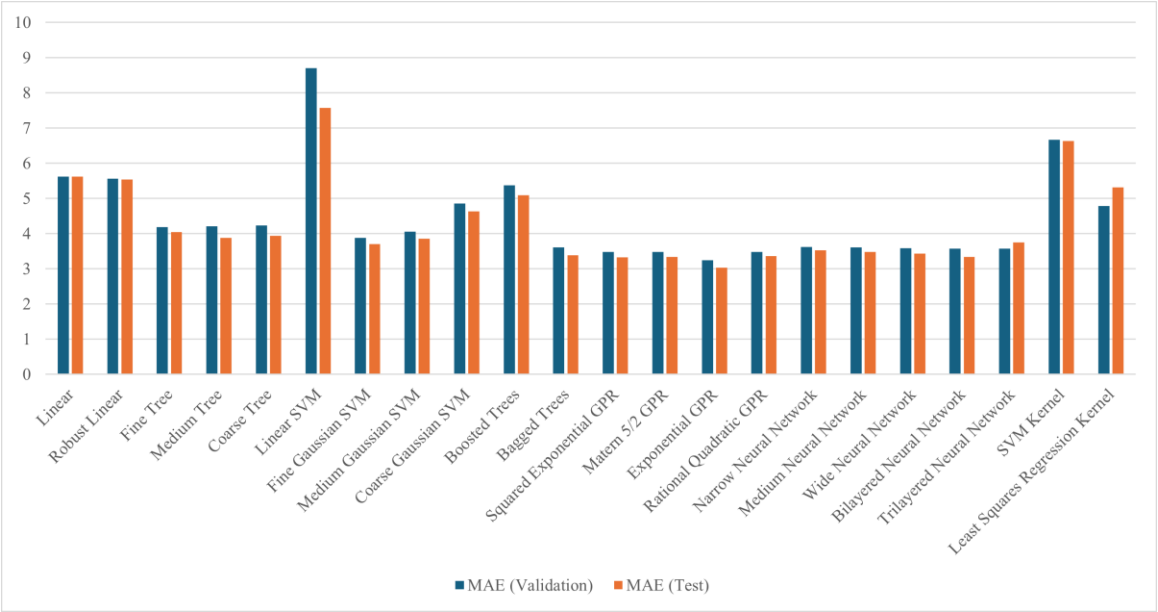

Figure 18 MAE for validation and test dataset.

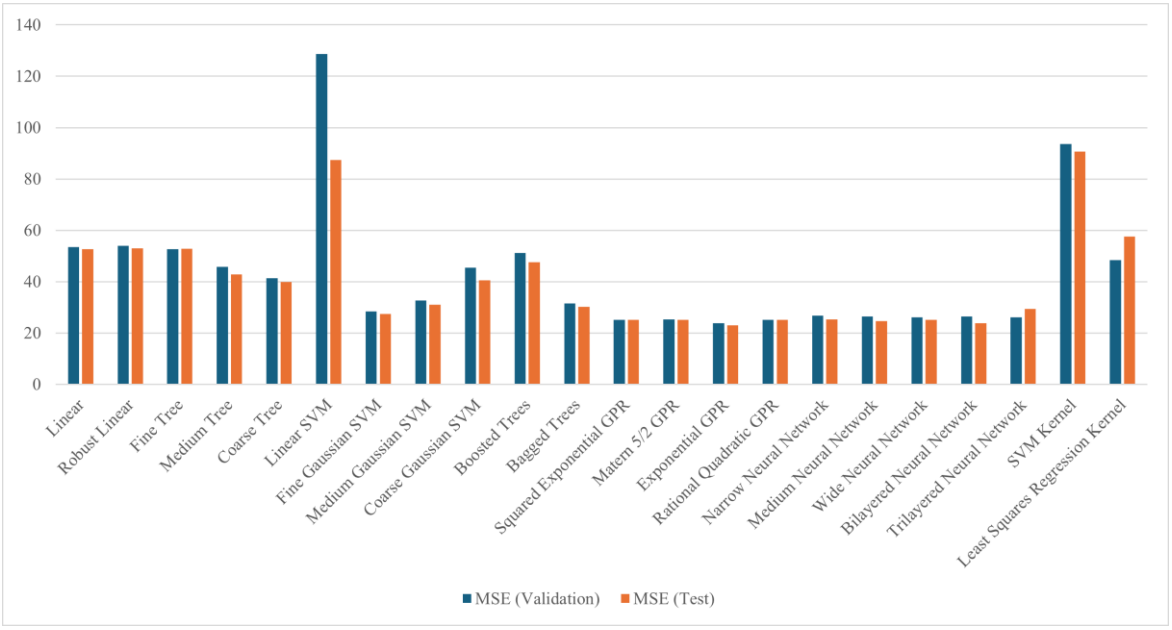

Figure 19 MSE for validation and test dataset.

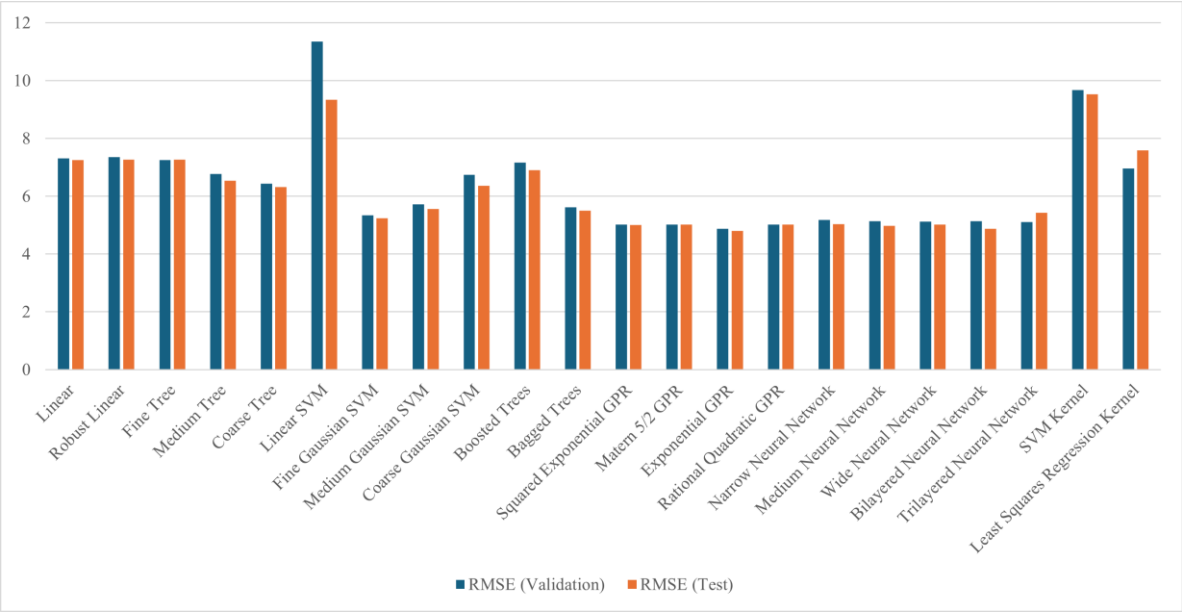

Figure 20 RMSE for validation and RMSE dataset.

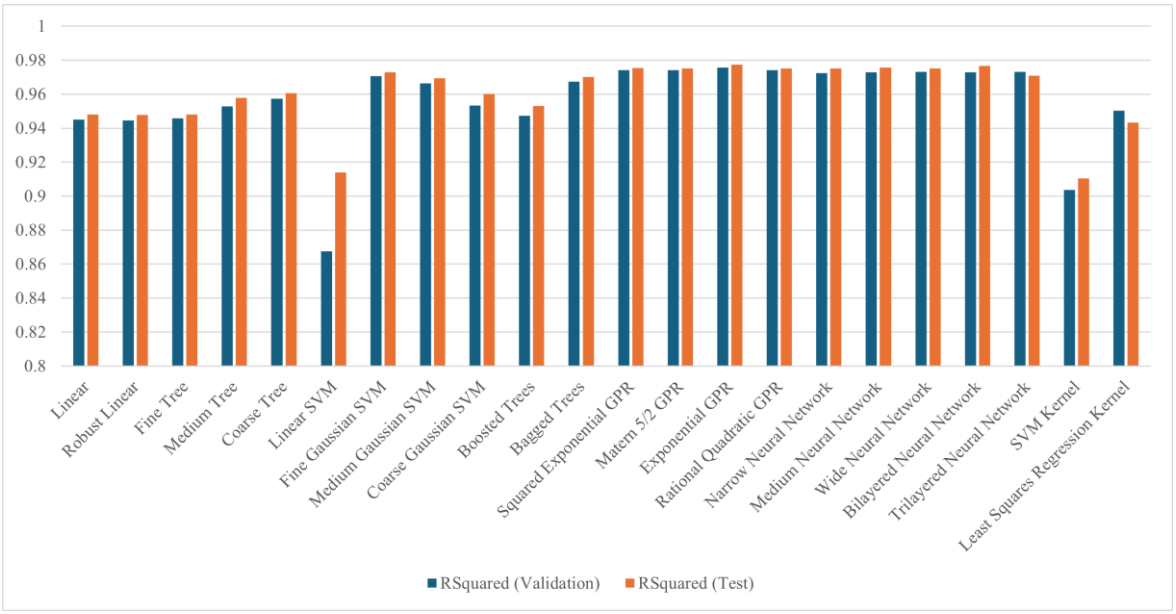

Figure 21 R Squared for validation and test dataset.

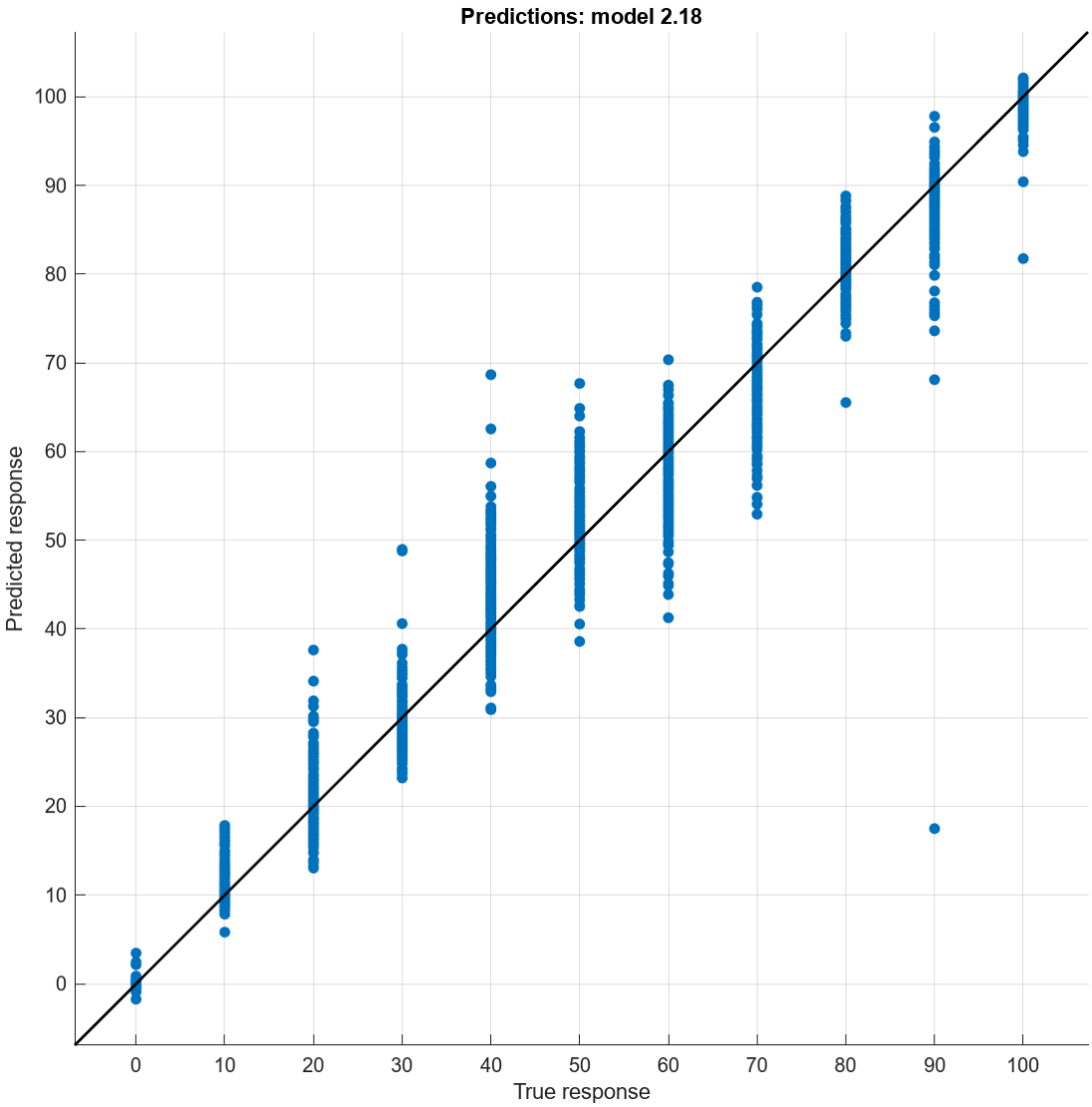

Figure 22 Predicted responses vs True responses.

The segmentation outcomes obtained through the application of the Fine Gaussian SVM are presented in Figure 23. The left-hand side (a) depicts the RGB image of the test sample, while the right-hand side (b) presents a color map of the segment results. From the figure, it can be observed that as the density of galena increases, the segments appear brighter, which suggests that the learning process was successfully performed.

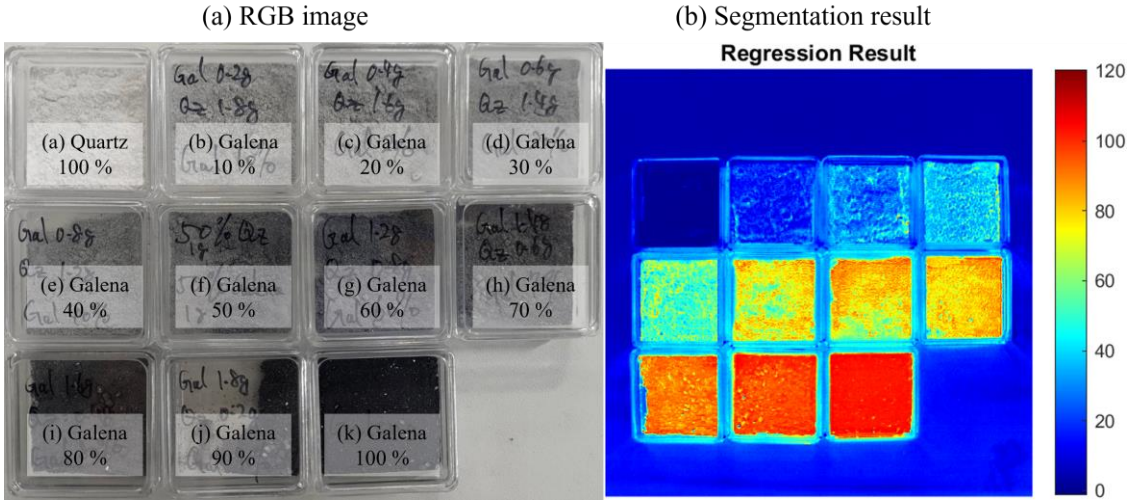

Figure 23 (a) Test sample's RGB image and (b) segmented result using Fine Gaussian SVM.

4.2.2. CNN

The same dataset was utilized for the purposes of learning and segmentation with a convolutional neural network (CNN). The learning curves are depicted in Figure 24. The top of the figure depicts the change in RMSE values for the RMSE training trials, while the orange figure below illustrates the loss. As illustrated in the accompanying figure, the root mean square error (RMSE) exhibited a convergence at 7.8472. Subsequently, the test hyperspectral data was subjected to segmentation using the trained convolutional neural network (CNN) model. The segmentation results are presented in Figure 25. As the concentration of galena increases, the image appears brighter, indicating that the regression analysis was performed successfully.

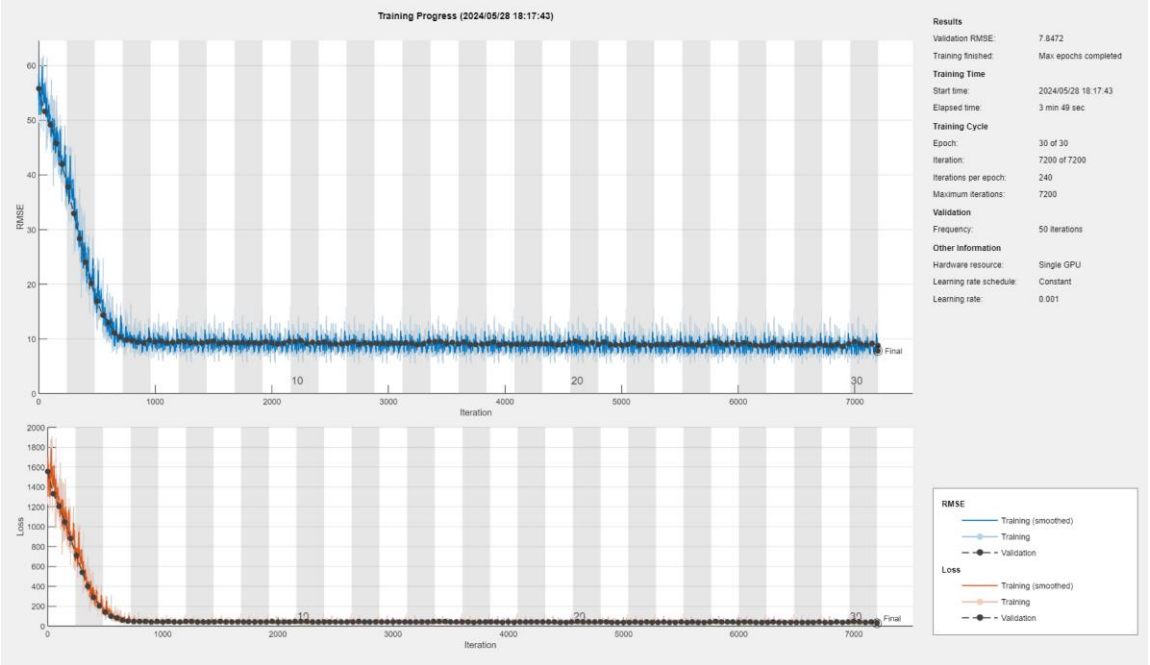

Figure 24 Training progress of regression CNN.

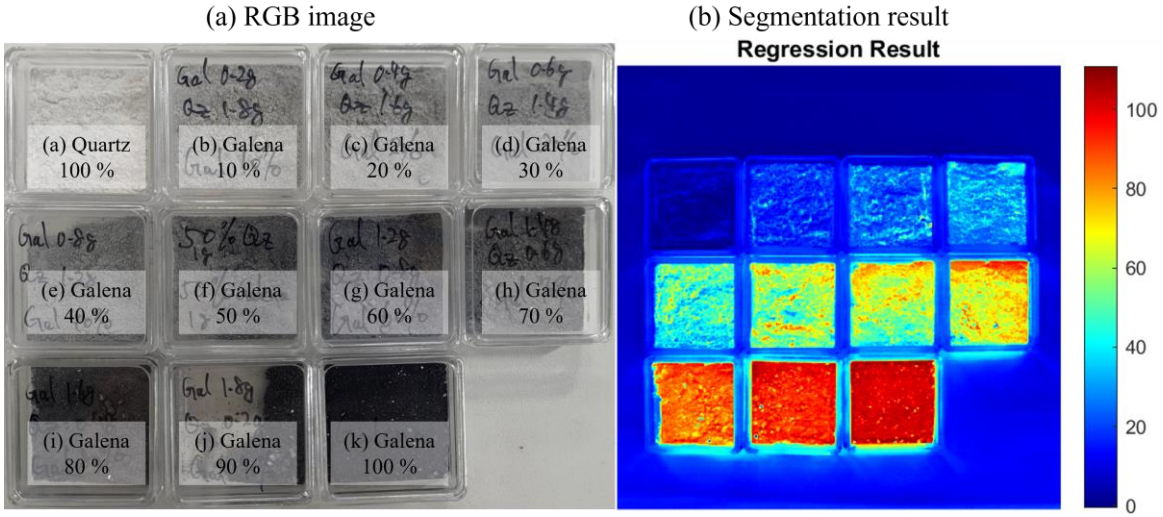

Figure 25 (a) Test sample's RGB image and (b) segmented result using CNN.

5. Limitations and Future Works

This application is designed to operate exclusively within the MATLAB programming environment and is not compatible with other programing languages. Additionally, at this juncture, the application does not facilitate the output of variables processed within the application, which may impede the user's capacity to obtain specific data. It is under consideration to implement the capability to output variables in the future. Furthermore, the number of hyperspectral data that can be read is contingent upon the computer performance utilized by the user. Consequently, the current implementation may be constrained for certain users when processing extensive data sets. Future enhancements may be made to more efficient data loading methods. It is anticipated that these enhancements will further enhance the user experience.

6. Conclusion

Hyperspectral data exhibits high identification performance due to its wavelength resolution. In particular, the field of mineral identification technology using hyperspectral data has made significant strides in recent years, largely due to the advent of artificial intelligence technology. However, the analysis of hyperspectral data by non-expert users is constrained by the data's distinctive characteristics, and the development of user-friendly analysis tools has been a long-standing aspiration.

The AI hyperspectral analysis tool APiS, developed by the authors, represents an innovative application based on a usability-oriented philosophy. It allows any MATLAB user to perform advanced machine learning processing of hyperspectral data with the simple touch of a button, without the necessity of writing code. In addition to its capabilities for classification and regression, APiS is also equipped to perform interactive labeling and preprocessing, which have previously been challenging to achieve.

Author Contributions:

Conceptualization, N.O(1)., N.O(2)., Y.O., and Y.K.;

Methodology, N.O(1)., K.T., B.B.S., A.G. and S.T.;

Software, N.O(1)., and N.O(2)..;

Validation, N.O(1)., K.A., and I.P.;

Formal analysis, N.O(1)., O.M., and A.G.;

Investigation, N.O(1)., O.M., and Y.O.;

Resources, S.T., K.A., I.P., and M.I.;

Data curation, N.O(1)., S.T., K.A., I.P., and M.I;

Writing-original draft, N.O(1).;

Writing-review & editing, N.O(1)., O.M., A.G., B.B.S., K.A., Y.O., and Y.K.;

Visualization, N.O(1).;

Supervision, Y.O., I.P., M.I., and Y.K.;

Project administration, Y.K.;

Funding acquisition, Y.K., N.O(1).;

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

Funding: This work was supported by JSPS KAKENHI Grant Number JP23KJ0029.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in the study are available from the corresponding author upon reasonable request.

Acknowledgments: The researchers would like to acknowledge the JSPS KAKENHI for funding this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] A. Plaza *et al.*, "Recent advances in techniques for hyperspectral image processing," *Remote Sens Environ*, vol. 113, no. SUPPL. 1, pp. S110–S122, Sep. 2009, doi: 10.1016/J.RSE.2007.07.028.

[2] E. Saralioglu, E. T. Gormus, and O. Gungor, "Hiperspektral Kaynastirma ile Mineral Tespiti," *2016 24th Signal Processing and Communication Application Conference, SIU 2016 - Proceedings*, pp. 1281–1284, Jun. 2016, doi: 10.1109/SIU.2016.7495981.

[3] B. Yousefi *et al.*, “Mineral identification in hyperspectral imaging using Sparse-PCA,” *https://doi.org/10.1117/12.2224393*, vol. 9861, pp. 312–322, May 2016, doi: 10.1117/12.2224393.

[4] F. J. A. Van Ruitenbeek *et al.*, “Mapping the wavelength position of deepest absorption features to explore mineral diversity in hyperspectral images,” *Planet Space Sci*, vol. 101, pp. 108–117, Oct. 2014, doi: 10.1016/J.PSS.2014.06.009.

[5] A. Guha, “Mineral exploration using hyperspectral data,” *Hyperspectral Remote Sensing: Theory and Applications*, pp. 293–318, Jan. 2020, doi: 10.1016/B978-0-08-102894-0.00012-7.

[6] S. Sudharsan, R. Hemalatha, and S. Radha, “A survey on hyperspectral imaging for mineral exploration using machine learning algorithms,” *2019 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2019*, pp. 206–212, Mar. 2019, doi: 10.1109/WISPNET45539.2019.9032740.

[7] F. A. Kruse, “Identification and mapping of minerals in drill core using hyperspectral image analysis of infrared reflectance spectra,” *Int J Remote Sens*, vol. 17, no. 9, pp. 1623–1632, 1996, doi: 10.1080/01431169608948728.

[8] N. Okada, Y. Maekawa, N. Owada, K. Haga, A. Shibayama, and Y. Kawamura, “Automated Identification of Mineral Types and Grain Size Using Hyperspectral Imaging and Deep Learning for Mineral Processing,” *Minerals*, vol. 10, no. 9, 2020, doi: 10.3390/min10090809.

[9] N. Okada, Y. Maekawa, N. Owada, K. Haga, A. Shibayama, and Y. Kawamura, “Classification of arsenic-bearing ores using hyperspectral imaging and deep learning in beneficiation processes,” *Journal of MMIJ*, vol. 137, no. 1, pp. 1–9, 2021, doi: 10.2473/journalofmmij.137.1.

[10] N. Okada *et al.*, “Advanced mineralogical classification and concentration estimation in mining with MATLAB-powered hyper-spectral imaging and machine learning,” *Int J Min Reclam Environ*, Jun. 2024, doi: 10.1080/17480930.2024.2360743.

[11] The MathWorks Inc., “Hyperspectral Image Processing - MATLAB & Simulink.” Accessed: Jun. 04, 2024. [Online]. Available: https://www.mathworks.com/help/images/hyperspectral-image-processing.html

[12] E. Arzuaga-Cruz *et al.*, “A MATLAB toolbox for hyperspectral image analysis,” *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 7, pp. 4839–4842, 2004, doi: 10.1109/IGARSS.2004.1370246.

[13] H. Feng *et al.*, “An integrated hyperspectral imaging and genome-wide association analysis platform provides spectral and genetic insights into the natural variation in rice,” *Scientific Reports 2017 7:1*, vol. 7, no. 1, pp. 1–10, Jun. 2017, doi: 10.1038/s41598-017-04668-8.

[14] A. Islam ElManawy, D. Sun, A. Abdalla, Y. Zhu, and H. Cen, “HSI-PP: A flexible open-source software for hyperspectral imaging-based plant phenotyping,” *Comput Electron Agric*, vol. 200, p. 107248, Sep. 2022, doi: 10.1016/J.COMPAG.2022.107248.

[15] N. Mobaraki and J. M. Amigo, “HYPER-Tools. A graphical user-friendly interface for hyperspectral image analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 172, pp. 174–187, Jan. 2018, doi: 10.1016/J.CHEMOLAB.2017.11.003.

[16] The MathWorks Inc., “Classification Learner App - MATLAB & Simulink.” Accessed: Jun. 04, 2024. [Online]. Available: https://www.mathworks.com/help/stats/classification-learner-app.html

[17] The MathWorks Inc., “Regression Learner App - MATLAB & Simulink.” Accessed: Jun. 04, 2024. [Online]. Available: https://www.mathworks.com/help/stats/regression-learner-app.html

[18] I. H. Sarker, “Machine Learning: Algorithms, Real-World Applications and Research Directions,” *SN Comput Sci*, vol. 2, no. 3, pp. 1–21, May 2021, doi: 10.1007/S42979-021-00592-X/FIGURES/11.

[19] N. Quadrianto and W. L. Buntine, “Regression,” *Encyclopedia of Machine Learning*, pp. 838–842, 2011, doi: 10.1007/978-0-387-30164-8_710.

[20] T. T. Teoh and Z. Rong, “Classification,” pp. 183–211, 2022, doi: 10.1007/978-981-16-8615-3_11.

[21] S. B. Kotsiantis, “Decision trees: A recent overview,” *Artif Intell Rev*, vol. 39, no. 4, pp. 261–283, Apr. 2013, doi: 10.1007/S10462-011-9272-4/METRICS.

[22] “Pattern Recognition and Machine Learning,” *Pattern Recognition and Machine Learning*, Dec. 2006, doi: 10.1007/978-0-387-45528-0.

[23] T. Cooke, “Two variations on fisher’s linear discriminant for pattern recognition,” *IEEE Trans Pattern Anal Mach Intell*, vol. 24, no. 2, pp. 268–273, Feb. 2002, doi: 10.1109/34.982904.

[24] T. G. Nick and K. M. Campbell, “Logistic Regression,” *Methods Mol Biol*, vol. 404, pp. 273–301, 2007, doi: 10.1007/978-1-59745-530-5_14.

[25] F. J. Yang, “An implementation of naive bayes classifier,” *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, pp. 301–306, Dec. 2018, doi: 10.1109/CSCI46756.2018.00065.

[26] S. Suthaharan, “Support Vector Machine,” pp. 207–235, 2016, doi: 10.1007/978-1-4899-7641-3_9.

[27] M. Pal and G. M. Foody, “Feature selection for classification of hyperspectral data by SVM,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2297–2307, May 2010, doi: 10.1109/TGRS.2009.2039484.

[28] N. Bhatia and Vandana, “Survey of Nearest Neighbor Techniques,” *IJCSIS) International Journal of Computer Science and Information Security*, vol. 8, no. 2, Jul. 2010, Accessed: Jun. 04, 2024. [Online]. Available: https://arxiv.org/abs/1007.0085v1

[29] M. Muja and D. G. Lowe, “Scalable nearest neighbor algorithms for high dimensional data,” *IEEE Trans Pattern Anal Mach Intell*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014, doi: 10.1109/TPAMI.2014.2321376.

[30] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, “Ensemble learning for data stream analysis: A survey,” *Information Fusion*, vol. 37, pp. 132–156, Sep. 2017, doi: 10.1016/J.INFFUS.2017.02.004.

[31] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 8, no. 4, p. e1249, Jul. 2018, doi: 10.1002/WIDM.1249.

[32] H. S. Stern, “Neural networks in applied statistics,” *Technometrics*, vol. 38, no. 3, pp. 205–214, 1996, doi: 10.1080/00401706.1996.10484497.

[33] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” *PeerJ Comput Sci*, vol. 7, pp. 1–24, Jul. 2021, doi: 10.7717/PEERJ-CS.623/SUPP-1.

Figure captions as a list