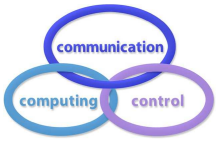
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 20, Issue: 5, Month: October, Year: 2025
Article Number: 6923, https://doi.org/10.15837/ijccc.2025.5.6923



CCC Publications

Advancing hyperspectral image analysis: harnessing machine learning for precision vegetation identification in urban areas

G.E Chanchí-Golondrino, M. A. Ospina-Alarcón, M. Saba

Gabriel Elías Chanchí-Golondrino
Faculty of Engineering
University of Cartagena
Av. del Consulado Cll. 30 #39B-192, Cartagena-Colombia
*Corresponding author: gchanchig@unicartagena.edu.co

Manuel Alejandro Ospina-Alarcón
Faculty of Engineering
University of Cartagena
Av. del Consulado Cll. 30 #39B-192, Cartagena-Colombia
mospinaa@unicartagena.edu.co

Manuel Saba
Faculty of Engineering
University of Cartagena
Av. del Consulado Cll. 30 #39B-192, Cartagena-Colombia
msaba@unicartagena.edu.co

Abstract

Exploiting the advantages inherent in machine learning methodologies, particularly supervised learning models tailored for conventional image classification, this study assesses their efficacy in delineating vegetation in hyperspectral images. The investigation adheres to a dataset comprising 400 spectral signatures, evenly divided between vegetation and non-vegetation categories, each characterized by 380 spectral bands. Five distinct models—KNN, decision trees, support vector machines, random forests, and logistic regression—are scrutinized for their performance, leveraging metrics derived from the confusion matrix and cross-validation. The research adopts a modified version of the CRISP-DM methodology, segmented into four phases: understanding the data and the domain, data preparation, modeling and evaluation, and model deployment. Throughout these phases, various open-source libraries such as spectral, scikit-learn, numpy, pandas, and matplotlib are employed. Results indicate that all five models achieve cross-validation accuracies surpassing 95% in vegetation pixel detection within hyperspectral images, with the KNN model exhibiting superior performance at 99.3% accuracy. Subsequently, the model with optimal performance is deployed on a hyperspectral image encompassing the Manga neighborhood in Cartagena, Colombia, comprising 2250000 pixels and 380 frequency bands, yielding highly effective vegetation pixel detection. This article introduces an approach intended to serve as a benchmark for the identification of diverse materials in hyperspectral images at both academic and industrial levels, utilizing open-source technologies.

Keywords: machine Learning method, urban vegetation, vegetation classification, supervised learning models.

1 Introduction

The field of vegetation mapping and monitoring has undergone a radical transformation with the advent of remote sensing technology. This powerful tool has empowered scientists and researchers to gain valuable insights into complex terrestrial environments [1, 2, 3, 4]. The capability to capture high-resolution images of the Earth's surface through satellites, aircraft, and drones has drastically altered the way we observe and analyze vegetation patterns, providing comprehensive knowledge about plant communities across various regions and biomes [1, 5, 6, 7, 8, 9, 10].

Despite outstanding advancements in remote sensing technology, the identification of vegetation in multi- and hyperspectral images remains a daunting challenge. This is due to the intricate spectral signatures of different plant species and the influence of environmental factors such as soil, water, and atmospheric conditions [11]. Traditional vegetation mapping methods utilizing remote sensing involve interpreting spectral indices like the Normalized Difference Vegetation Index (NDVI). This index measures vegetation cover based on the contrast between the red and near-infrared bands of the electromagnetic spectrum, making operation between bands [9, 12, 13, 14, 15, 16, 17, 18, 19].

Additionally, correlation methods between curves have emerged as an effective tool for comparing spectral signatures of hyperspectral image pixels and classifying vegetation. Spectral signatures represent the reflectance values of an object or material across the electromagnetic spectrum. In the context of vegetation mapping, these signatures help identify plant species or vegetation types based on their unique spectral properties [20] . Common correlation methods used for vegetation mapping include direct correlation, cosine similarity, normalized Euclidean distance, Bray–Curtis distance, Pearson correlation and spectral angle mapper, among others [21, 22, 23].

Pearson's correlation coefficient quantifies the linear relationship between two spectral signatures, while the spectral angle mapper measures the angular similarity, ranging from 0 to 1, with higher values indicating more spectral similarity [22, 24, 25]. However, despite their effectiveness, these methods have limitations, such as the need for manual interpretation and the limited spatial resolution of satellite images. They are also time-consuming and costly, requiring extensive field surveys and ground truth data for calibration and validation [11, 26, 27].

To address these challenges, machine learning techniques have emerged as a promising approach for vegetation mapping and classification. These algorithms offer a cost-effective and accurate solution for remote sensing applications, particularly in developing countries. Machine learning can analyze vast amounts of hyperspectral data, identify intricate patterns, and provide automated and objective vegetation cover detection. The increasing availability of large hyperspectral datasets and the development of advanced algorithms and computational tools have fueled the rise of machine learning applications in vegetation mapping [28, 29, 30].

Various machine learning techniques, including artificial neural networks [31, 32, 33], decision trees [22, 34], support vector machines [35], and random forests [36, 37], have shown promising results in identifying and classifying vegetation cover from hyperspectral images. They offer a more robust and reliable solution compared to traditional methods. Moreover, machine learning techniques can be computationally effective and easily implementable, making them particularly advantageous for regions with limited resources and developing countries. Accordingly, it is necessary in the context of research centers in developing countries, the identification of open-source tools and technologies that not only enable the processing of hyperspectral images, but also the adjustment, evaluation and deployment of supervised learning methods for the detection of materials in this type of images.

In this work we propose as a contribution, the implementation, adjustment, evaluation and comparison of supervised learning models for the identification of vegetation in hyperspectral images from the use of open-source technologies. For the fitting and evaluation of the models, a dataset was conformed with 400 spectral signatures each with 380 reflectance bands (from 400 nm to 2400 nm), which were obtained from samples taken from a reference hyperspectral image with 2250000 pixels of 380 bands, which corresponds to the Manga neighborhood of the city of Cartagena-Colombia. The best model obtained from the confusion matrix and from cross-validation was deployed and/or validated on the reference hyperspectral image in order to obtain the percentage of vegetation available in the Manga neighborhood of Cartagena. The results obtained in this research are intended to serve as a basis for the implementation of environmental monitoring systems, which allow, for example, the iden-

tification of areas with vegetation cover to evaluate air quality and its capacity to absorb atmospheric pollutants. Similarly, the current research aims to serve as a reference regarding the extrapolation of these methods and technologies in the detection of other types of materials in hyperspectral images by research centers and companies. In this regard, the advantages of open-source technologies are not only associated with licensing costs but also with the potential to enhance the effectiveness and efficiency of detection methods by hybridizing different approaches.

The remainder of the article is organized as follows: Section 2 outlines the methodological phases considered in the development of this study. Section 3 describes the results and discussion, where, in terms of results, the fitting and comparative evaluation of the five machine learning models considered are presented, along with the deployment of the best vegetation detection model on a hyperspectral image of the Manga neighborhood in the city of Cartagena, Colombia. Finally, Section 4 presents the conclusions and future work derived from this research.

2 Methodology

The methodology employed encompasses a multifaceted approach designed to harness the power of various machine learning techniques for the accurate and robust identification of vegetation in hyperspectral imagery. Recognizing the critical role of accurate vegetation detection in applications ranging from agriculture to environmental monitoring, this research endeavors to address the complexity of hyperspectral data and its potential for precise classification.

For the development of the current research, an adaptation of the methodology for data science projects known as CRISP-DM was conducted [38, 39]. This adaptation defined four phases, namely: understanding of the business and data, data preparation, modeling and evaluation, and finally, model deployment (see Figure 1).

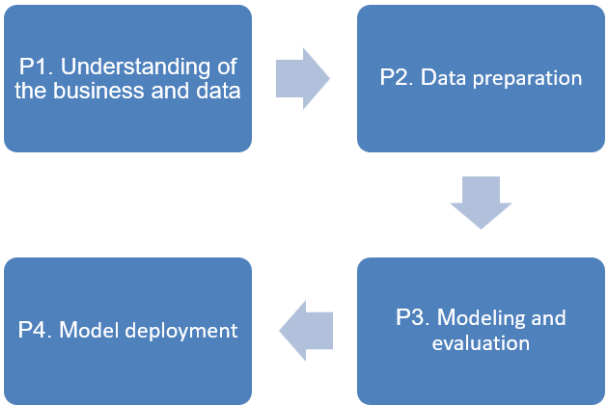


Figure 1: Methodology considered (Source)

In Phase 1 of the methodology, the hyperspectral data undergoes meticulous preprocessing, including spectral dimensionality reduction, noise reduction, and spatial contextual analysis through radiometric, geometric and atmospheric corrections, with the HySpex RAD, Rese PARGE and Rese DROACOR software, respectively. These steps aim to enhance the quality and interpretability of the data, facilitating subsequent machine learning operations. Then, a dataset of hyperspectral signatures was conformed, comprising a total of 200 vegetation signatures and 200 non-vegetation signatures (water, roads, containers, vehicles, etc.), each with 380 spectral bands. These signatures were extracted from a hyperspectral image of the Manga neighborhood in the city of Cartagena de Indias, which includes a total of 2,250,000 pixels of 0.4m x 0.4m, each with 380 bands. The elevated level of ground precision encountered herein, though atypical in extant literature, proves imperative in the investigation of urban environments. The image of the Manga neighborhood was constructed by merging strips generated by an airborne sensor. To conform this dataset, a visual inspection was conducted on the RGB image of the Manga neighborhood. Through this inspection, the coordinates of vegetation and non-vegetation pixels were identified, from which, using open-source tools, information for the

400 pixels and their 380 associated bands was obtained in a spreadsheet. Python libraries, including spectral, numpy, pandas, and matplotlib, were used for the acquisition and storage of samples. The spectral library allows access to pixel information in the image and retrieves each pixel as a numpy array. The numpy library enables operations on multidimensional arrays. The pandas library facilitates the reading and processing of various file types, including spreadsheets. Additionally, the matplotlib library enables the generation of graphs based on different data structures.

From the dataset conformed in Phase 1 of the methodology, Phase 2 initially involved the normalization (between 0 and 1) of reflectance values for each of the 400 pixels in the dataset. This was done to enhance the efficiency of the algorithmic operations of the machine learning models contrasted in this study. Once the reflectance values were normalized, the dataset of 400 rows and 380 columns was divided into training and testing sets, using a 75% split for training the models and the remaining 25% for evaluation using accuracy metrics. This process utilized functionalities provided by the sklearn library in Python. In Phase 3 of the methodology, five supervised learning models (KNN, decision trees, support vector machines, random forests, and logistic regression) were initially selected to be fitted using the training set and evaluated through metrics derived from the confusion matrix (precision, recall, and F1 score). Each method has its own capacity to capture different aspects of the data. KNN leverages proximity-based classification, SVM excels in finding nonlinear boundaries, Logistic Regression provides probabilistic outputs, Decision Trees offer interpretability, and Random Forests harness ensemble learning to enhance classification accuracy.

Additionally, cross-validation was employed to determine the best model for vegetation detection in hyperspectral images. Model performance is evaluated using cross-validation techniques to gauge accuracy and precision. Both the implementation and evaluation of the models were conducted using the advantages provided by the sklearn library in Python.

By combining these methodological elements, the authors endeavor to offer a comprehensive solution for vegetation identification in hyperspectral imagery. This multifaceted approach leverages the unique strengths of each machine learning algorithm to provide reliable, interpretable, and actionable insights for applications ranging from agriculture and forestry management to environmental conservation and land use planning. The subsequent sections of this paper delve into the specific details and outcomes of each method's application, contributing to a deeper understanding of their individual contributions and collective synergy in the realm of hyperspectral vegetation identification.

2.1 The K-Nearest Neighbors (K-NN) model

The K-Nearest Neighbors (K-NN) model is a supervised learning algorithm used for both classification and regression problems in the field of machine learning. Its approach is based on the principle that similar objects tend to be close in a multidimensional space [40, 41, 42]

The core principle behind the K-NN model is that similar objects tend to be close to each other in a multidimensional space. In other words, if most of the points nearest to a test point belong to a certain class, then that test point will be classified into that same class. The value of K is a crucial hyperparameter in K-NN. It represents the number of nearest neighbors to consider when making a prediction. Choosing an appropriate value for K is essential because a value that is too small can lead to overfitting (noise), while a value that is too large can lead to underfitting (bias). There is no specific equation for choosing K, but it is generally chosen by methods such as cross-validation or the use of rules of thumb [43].

To determine which are the "nearest neighbors," a distance function is used, commonly the Euclidean distance in problems with numerical data (see Eq 1), where xi and yi are the values of the characteristics at the two points being compared. However, for categorical features or other data types, you may need to use a different distance metric. In the case of classification, once the K nearest neighbors are identified, the majority class among these neighbors is taken as the prediction class for the test point. In regression, instead of taking the majority class, the average (or some other metric) of the target values of the K nearest neighbors is taken as the prediction value for the test point [44].

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{1}$$

It's important to perform feature scaling in K-NN since features with vastly different scales can have unequal impacts on the distance calculation. Scaling helps ensure that all features have equal weight in distance computation. K-NN can be sensitive to noise and outliers in the data as it relies on point proximity. Outliers can significantly influence classification decisions or regression estimates [45].

K-NN can be computationally expensive on large datasets because it involves calculating the distance between the test point and all training points. However, there are variants and techniques to speed up calculations, such as using data structures like KD-Trees or approximate algorithms like Local Sensitive Hashing (LSH) [45]. It should be noted that although KNN is a basic technique, it has proven to be effective in the classification of conventional images, offering a greater advantage in the context of hyperspectral image classification due to the large amount of relevant information provided by the reflectance bands [46].

2.2 Support Vector Machines (SVM)

The core idea behind Support Vector Machines (SVM) is to find the hyperplane that best separates two classes in a high-dimensional feature space. This hyperplane is chosen in such a way that it maximizes the margin between the two classes, which helps improve the model's generalization performance [47, 48]. The mathematical representation of the hyperplane in an SVM can be expressed as Eq 2 [49].

$$W \cdot x + b = 0 \tag{2}$$

Where, W is the weight vector that defines the orientation of the hyperplane, x represents the input features and b is the bias term, which shifts the hyperplane away from the origin. The goal of an SVM is to find the optimal W and b values that maximize the margin while minimizing classification errors. This is typically formulated as a constrained optimization problem. The margin is computed as the distance between the hyperplane and the closest data points, which are called support vectors. The optimization problem can be expressed as Eqs 3 and 4 [50, 51].

$$\frac{1}{2}\|W\|^2 \tag{3}$$

Subject to:

$$y_i(W \cdot x_i + b) \geq 1 \quad \text{for all training samples} \quad (x_i, y_i) \tag{4}$$

Where, $\|W\|$ represents the Euclidean norm of the weight vector W, y_i is the class label (+1 or -1) for the i-th training sample.

The above optimization problem is often solved using techniques like the Lagrange multiplier method to find the optimal values of W and b. Once these values are determined, the SVM can make predictions for new data points by evaluating the sign of the left-hand side of Eq 2 [48].

In practice, SVM's can be extended to handle non-linearly separable data by using kernel functions. The kernel trick allows SVM's to map the input features into a higher-dimensional space where the data may become linearly separable. Common kernel functions include the linear kernel, polynomial kernel, and radial basis function (RBF) kernel [49].

2.3 Decision Tree Model

Decision trees are a popular machine learning technique for classification tasks, including hyperspectral image classification [52, 53]. In this context, each pixel in a hyperspectral image is treated as a data point with multiple spectral bands (typically hundreds), and the goal is to assign a class label to each pixel based on its spectral signature [53].

Entropy (H) measures the impurity or disorder of a set of data points. In the context of a decision tree, it's used to quantify the uncertainty of class labels. The formula for entropy is given to Eq 5 [54].

$$H(S) = -\sum_i p(C_i)\log_2(p(C_i)) \tag{5}$$

Where H(S) is the entropy of set S, Ci represents each unique class in the dataset, p(Ci) is the proportion of data points in class Ci within set S. On the other hand, information gain (IG) measures the reduction in entropy achieved by splitting a dataset based on a particular attribute (spectral band in this case). The formula for IG is represented by Eq 6 [55].

$$IG(S,A) = H(S) - \sum_{v\in \text{values}(A)} \frac{|S_v|}{|S|}\cdot H(S_v) \tag{6}$$

Where, IG(S, A) is the information gain achieved by splitting set S using attribute A, H(S) is the entropy of set S, Sv represents subsets of S created by splitting S based on the values of attribute A. Decision trees are constructed recursively by selecting the attribute (spectral band) that maximizes information gain at each node. The tree is grown until certain stopping criteria are met, such as a maximum depth or a minimum number of samples per leaf node. Leaf nodes in the decision tree represent class labels. When a hyperspectral pixel reaches a leaf node, it is assigned the class label associated with that leaf [56, 57]. To avoid overfitting, pruning techniques can be applied to simplify the decision tree by removing branches that provide little predictive power [53, 54, 55].

2.4 Logistic Regression

Logistic Regression (LR) is a commonly used machine learning algorithm for binary classification tasks [58, 59, 60], which means it is used to categorize data into one of two classes, such as "Yes" or "No," "True" or "False," or, in the present case, "vegetation" or "non-vegetation" in hyperspectral images. LR models the probability that a given input belongs to a particular class. In the present case study, it predicts the probability that a pixel in the hyperspectral image represents vegetation (class 1) or non-vegetation (class 0). Thus, although logistic regression may be assumed to be a regression method based on its name, it is actually a statistical classification method used to model the probability of a binary outcome based on one or more independent variables (vegetation and non-vegetation pixels). It employs the logistic function to transform any real number into a probability value between 0 and 1, enabling the classification of data into discrete classes. This classification is performed using the sigmoidal function, which maps input values to probabilities, facilitating decision-making based on a predefined threshold [61, 62].

The logistic function, often denoted as $\sigma(z)$, maps any input value z to a value between 0 and 1. It is used to model the probability of the input belonging to class 1. The formula for the LR is shown below in Eq. 7.

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{7}$$

Where, z is a linear combination of input features and model parameters,

$$z = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n.$$

w_0 is the bias or intercept term, $w_1, w_2, \ldots, w_n$ are the weights associated with each feature $x_1, x_2, \ldots, x_n$, e is the base of the natural logarithm (approximately 2.71828). The logistic regression model is defined as in Eq. 8:

$$P(Y = 1 \mid X) = \sigma(w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n) \tag{8}$$

$P(Y = 1 \mid X)$ represents the probability that the input X belongs to class 1 (vegetation), X is the input vector containing spectral data for a pixel. The σ function is the logistic (sigmoid) function, $w_0, w_1, w_2, \ldots, w_n$ are the model parameters to be learned from the training data.

To train a logistic regression model for vegetation identification, it is typically used a labeled dataset of hyperspectral images [58, 63]. Techniques like gradient descent or optimization algorithms are used to learn the optimal weights $w_0, w_1, w_2, \ldots, w_n$ that minimize the logistic loss function. After

training, the learned weights can be used to make predictions on new hyperspectral images. For each pixel's spectral data, it is computed z using the learned weights and applying the sigmoid function to obtain the probability of vegetation. A threshold (e.g., 0.5) is set to classify the pixel as vegetation or non-vegetation based on this probability.

2.5 Random Forest

Random Forests involve a combination of decision trees, and the mathematical details for Random Forests primarily revolve around how these decision trees are constructed and aggregated [29, 30, 64, 65, 66]. Random Forests use bootstrapped samples from the training data to create diverse subsets for training each decision tree. Given a dataset with 'N' samples, a bootstrapped sample of size 'N' is created by randomly selecting 'N' samples from the dataset with replacement. The probability of a sample being selected is 1/N. This process is represented mathematically as Eq 9.

$$P(\text{sample } i \text{ is in the bootstrapped sample}) = \frac{1}{N} \tag{9}$$

Each decision tree in a Random Forest is constructed using a recursive binary splitting process. The goal is to find the best feature and threshold to split the data into two subsets at each node. The Gini impurity or information gain is commonly used to measure the quality of a split. For binary classification, the Gini impurity formula is presented in Eq 10.

$$Gini(D) = 1 - P(\text{class } 1)^2 - P(\text{class } 0)^2 \tag{10}$$

Where D is a dataset at a particular node, $P(\text{class } 1)$ is the proportion of samples in class 1 in dataset D, and $P(\text{class } 0)$ is the proportion of samples in class 0 in dataset D. Decision trees aim to minimize impurity, and the best split is chosen based on this criterion. To introduce randomness, Random Forests consider only a random subset of features (spectral bands in the context of hyperspectral imagery) when making splitting decisions at each node of a decision tree. The number of features to consider at each split is determined by a hyperparameter m, typically referred to as "max_features". For example, if m is set to the square root of the total number of features, N, then $\sqrt{N}$ features are randomly chosen at each split.

Once all decision trees are constructed, predictions from individual trees are aggregated to make the final prediction. In classification tasks, this is typically done through majority voting. The class that receives the most votes among all decision trees is considered the final prediction. For example, if you have K decision trees, and each tree predicts the class for a given input X as $C_1, C_2, \ldots, C_K$, then the final prediction is: Final Prediction for $X = \text{Mode}(C_1, C_2, \ldots, C_K)\ (X)$. Where "Mode" represents the most frequently occurring class among the predictions. These mathematical components describe how Random Forests utilize bootstrapping, decision trees, feature randomness, and ensemble aggregation to make predictions on new data. Random Forests' strength lies in the combination of multiple decision trees, which reduces overfitting and improves generalization,

Finally, in Phase 4 of the methodology, the deployment of the best model identified from the conducted comparison was carried out. This model was applied to detect vegetation areas on the 1500 x 1500 pixel image with 380 bands, which was used to obtain the samples for compiling the dataset in this research. The application of the model allowed determining the percentage of vegetation present in the image. Consequently, the use of the proposed model will enable Colombian environmental authorities to monitor changes in vegetation in the city of Cartagena over time.

3 Results and Discussion

Initially, for the conformation of the dataset, a total of 400 spectral signatures were sampled from pixels of vegetation and non-vegetation in a hyperspectral image of the Manga neighborhood in the city of Cartagena. This image was structured by integrating different image strips captured by an aerial remote sensor. In Figure 2, pixels of vegetation, whose spectral signatures were used for dataset formation, are depicted in green, while pixels of materials other than vegetation, whose

spectral signatures were employed in structuring the dataset, are presented in red. A total of 200 samples of vegetation and 200 samples of non-vegetation were taken to ensure a balanced dataset that does not bias the models' predictions. The points shown in Figure 2 were automatically drawn on the image using the OpenCV library in Python, iterating through an array containing the coordinates of the spectral signatures of the sample vegetation and non-vegetation pixels via a for loop.

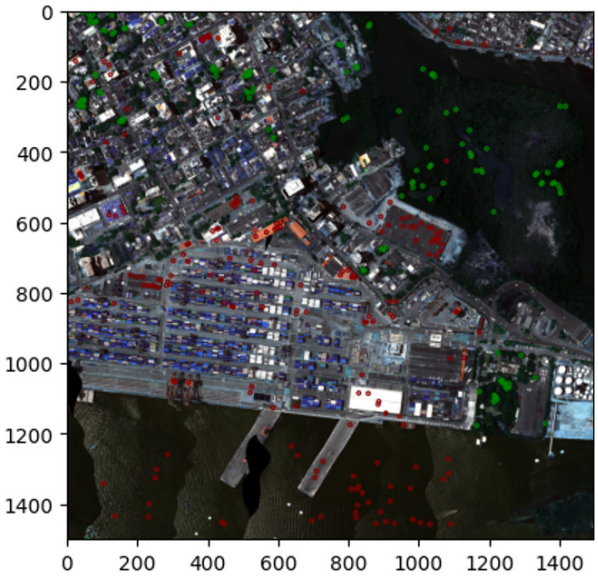


Figure 2: Sample of pixels considered in the dataset

Accordingly, the obtained dataset consists of a total of 400 rows and 380 columns corresponding to the frequency bands, with an additional column (type) where the label distinguishing a vegetation pixel from a non-vegetation pixel was added (see Figure 3). Thus, each row of the dataset includes the reflectance values presented by each pixel in the 380 bands (not normalized).

	0	1	2	3	4	5	6	7	8	9	...	371	372	373	374	375	376	377	378	379	type
0	262	265	272	278	284	288	299	308	313	324	...	736	924	1131	1249	1317	1350	1280	1238	0	1
1	256	269	276	287	295	309	322	329	326	326	...	809	843	933	967	1034	1104	1187	1212	0	1
2	308	316	325	340	355	368	384	395	401	411	...	959	955	924	830	748	662	612	631	0	1
3	362	358	353	349	347	344	344	347	349	365	...	971	1017	1043	1030	1029	1003	1013	1022	0	1
4	293	312	328	340	352	374	393	406	401	407	...	935	946	947	888	813	781	774	799	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	971	1045	1109	1159	1214	1283	1354	1410	1414	1456	...	2268	2207	2157	2095	2008	1949	1934	1820	0	0
396	932	993	1034	1088	1145	1205	1268	1322	1325	1368	...	1742	1698	1719	1797	1820	1798	1848	1854	0	0
397	679	714	734	759	787	820	860	896	899	928	...	1268	1349	1469	1606	1565	1396	1171	856	0	0
398	645	669	698	731	759	793	832	869	874	902	...	1731	1888	1903	1919	1765	1533	1276	1008	0	0
399	599	633	657	685	712	742	772	798	797	827	...	1485	1438	1478	1534	1671	1799	1962	2104	0	0

Figure 3: Conformed dataset

Similarly, it is worth highlighting that each row of the dataset corresponds to a spectral signature, either of vegetation or non-vegetation, corresponding to a pixel in the image. Accordingly, Figure 4 separately displays the 200 spectral signatures of vegetation and non-vegetation that constitute the dataset used in this article. These signatures were used to fit the machine learning models considered. It should be noted that the sample vegetation pixels all have a characteristic spectral signature specific to this type of material (with differences in reflectance amplitude), making it distinct from the spectral signature of other pixels. This distinction ensures that machine learning models are trained with sample pixels that are not mixed with other materials [67].

It can be observed from Figure 4 that, although the spectral signatures associated with vegetation pixels exhibit varying magnitudes in reflectance, the shape of the different curves is similar. This similarity can facilitate the algorithmic operations performed by the models. Once the dataset and the representation of each row in the model were understood, the data associated with the spectral bands were normalized to the range of 0 to 1. This normalization aims to make the comparisons between models more efficient. It is crucial to ensure that the different magnitudes of the features

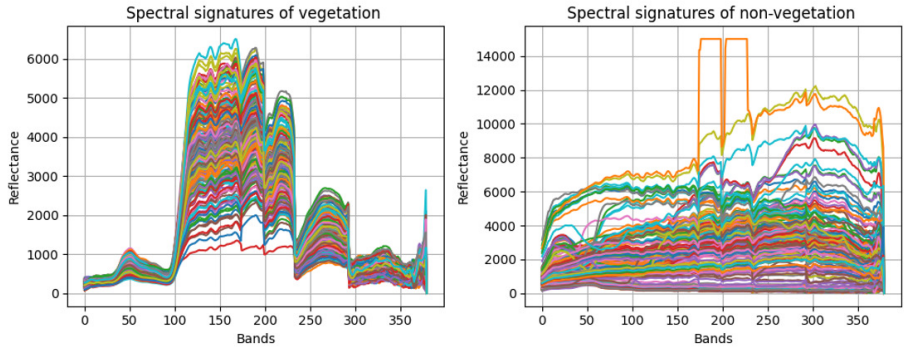


Figure 4: Spectral signatures of vegetation and non-vegetation pixels.

do not bias the performance of the models, allowing for a fair evaluation. Subsequently, the dataset was split into training and validation sets, using 75% for training and 25% for validation. This approach ensures the assessment of model performance on unseen data and provides a more realistic measure of their ability to generalize to new instances. From the training set, the fitting of the 5 machine learning models considered in this article was carried out: KNN, decision trees, support vector machines, random forests, and logistic regression.

Once the models were fitted, the validation process was carried out, evaluating their performance using the test set. In this context, the confusion matrix emerges as an essential tool for a detailed analysis of the effectiveness of each model. This matrix provides a comprehensive view of precision and classification ability, breaking down the number of true positives, true negatives, false positives, and false negatives. Its importance lies in its capacity to reveal both the strengths and potential weaknesses of the models, enabling informed decision-making in the continuous improvement of system performance. Similarly, the cross-validation approach was employed, allowing robust validation of the models by varying the training and test sets within the same dataset.

Thus, regarding the KNN model, through the methods provided by the scikit-learn library, the confusion matrix was obtained, as presented in Figure 5. The confusion matrix obtained for the KNN model (k-nearest neighbors) with 3 neighbors reveals an adequate performance in classifying pixels of vegetation in hyperspectral images, focusing on the distinction between vegetation and non-vegetation. The first row of the matrix reflects that 44 pixels were correctly classified as vegetation, representing true positives. In the second row, it is observed that the 56 non-vegetative pixels were accurately identified, constituting true negatives. Thus, the absence of false positives and false negatives in this matrix indicates an optimal level of precision in the KNN model with 3 neighbors. It is worth mentioning that within the KNN model, it is necessary to specify an odd number of neighbors (3, 5, 7, 9, etc.) so that, within the neighborhood formed by the model, the algorithm selects among the number of categories to classify, in this case, two: vegetation pixels and non-vegetation pixels. A choice of three neighbors was made since increasing the number of neighbors did not improve the model's accuracy, and this number ensures lower computational cost.

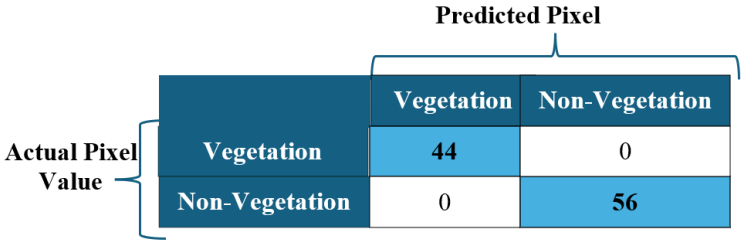


Figure 5: Confusion matrix for KNN.

From the confusion matrix presented in Figure 5, the following metrics derived from it were obtained for the KNN model: precision, recall, and F1-score, which are presented in Table 1. Similarly, in Table 1, an additional presentation of the precision metric obtained through the 5-fold cross-validation

Table 1: Metrics obtained for the KNN model.

Metric	Results
Precision	1.0
Recall	1.0
F1-Score	1.0
Accuracy obtained by cross-validation	0.9933

method is provided.

According to the results obtained in Table 1, it is evident that the precision percentage of 100% indicates that all instances in the validation set classified by the KNN model are truly positive, meaning they correspond to vegetation pixels. Similarly, concerning the recall metric, which evaluates the model's sensitivity to accurately distinguish positive instances (vegetation pixels), it is noteworthy that KNN has achieved a value of 1. This indicates that the model successfully identifies all positive instances in the dataset, i.e., vegetation pixels, without leaving any instance unclassified. This aspect is crucial to minimize the detection of false negatives, representing non-vegetation pixels incorrectly identified as vegetation pixels. Regarding the F1-score metric, achieving a percentage value of 100%, and considering that this metric is calculated from the harmonic mean between precision and recall, it can be concluded that there is an optimal balance between both metrics. In other words, both metrics obtained a perfect score. Concerning the F1-score metric, which reaches a value of 100%, it is worth noting that this metric is calculated as the harmonic mean between precision and recall. This result suggests an optimal balance between both metrics, indicating that both precision and recall have achieved a perfect score. Finally, when determining the model's precision through 5-fold cross-validation, an average precision percentage of 99.33% was obtained. This suggests consistency in the model's performance and its high effectiveness when using the model with different validation data.

In the case of the decision tree model, the confusion matrix obtained through the advantages provided by the scikit-learn library is presented in Figure 6. The confusion matrix obtained for the decision tree model in the classification of vegetation pixels in hyperspectral images reveals a good performance in differentiating between vegetation and non-vegetation pixels. In the first row of the matrix, it is highlighted that 44 pixels were correctly classified as vegetation, representing true positives. However, in the second row, it is evident that 2 non-vegetation pixels were incorrectly classified as vegetation, constituting false negatives. Additionally, 54 non-vegetation pixels were correctly identified, representing true negatives. Although the presence of false negatives indicates instances of error in classification, the absence of false positives suggests an effective ability of the model to identify vegetation pixels. Thus, the decision tree model demonstrates good performance in the classification of vegetation and non-vegetation pixels, with an emphasis on minimizing false positives.

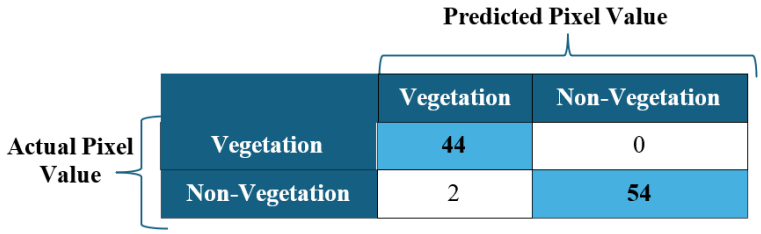


Figure 6: Confusion matrix for Decision Trees.

Based on the confusion matrix shown in Figure 6, precision, recall, and F1-score metrics were derived, along with the precision metric obtained through the 5-fold cross-validation model. The results for these metrics by the decision tree model are specified in Table 2.

According to the results obtained for the decision tree model presented in Table 2, an outstanding performance in the classification of vegetation pixels is evident, achieving a percentage of 98% in precision, recall, and F1-Score metrics. The 98% precision represents a high and nearly perfect proportion of positive instances correctly classified as positive (vegetation pixels) among the total positive and negative instances. Similarly, regarding the recall metric, set at 98%, it indicates the model's ability to

Table 2: Metrics obtained for the Decision Trees model.

Metric	Results
Precision	0.98
Recall	0.98
F1-Score	0.98
Accuracy obtained by cross-validation	0.986

identify all positive instances (vegetation pixels) present in the data. Furthermore, the 98% F1-score highlights a balance between positive classification ability and the minimization of false positives and false negatives. Similarly, the 5-fold cross-validation, with a value of 98.6%, indicates the consistency of the model's performance with different subsets of data.

In regards to the support vector machine model under the polynomial approach, the confusion matrix presented in Figure 7 was obtained using the functionalities provided by the scikit-learn library. The confusion matrix obtained for the support vector machine model in the classification of vegetation pixels in hyperspectral images reveals good performance in differentiating vegetation pixels from non-vegetation pixels. In this way, in the first row of the matrix, it is noteworthy that 44 instances were correctly classified as vegetation, representing true positives. Despite this, in the second row, it is evident that 7 non-vegetation pixels were incorrectly classified as vegetation, representing false negatives. Similarly, 49 non-vegetation pixels were correctly identified, representing true negatives. Although the presence of false negatives indicates errors in the model, the absence of false positives suggests the model's capability to identify vegetation pixels. Thus, the support vector machine model demonstrates good performance in the classification of vegetation and non-vegetation pixels, with an emphasis on minimizing false positives.

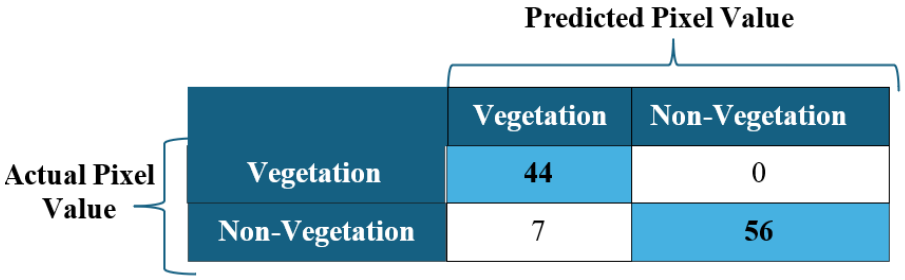


Figure 7: Confusion matrix for support vector machine.

Based on the confusion matrix presented in Figure 7, precision, recall, and F1-score metrics were obtained, along with the precision metric derived from the 5-fold cross-validation model. The results for these metrics are specified in Table 3. According to the results presented in Table 3 for the support vector machine model, it is possible to appreciate good performance, although inferior to the KNN and decision tree models, in the classification of vegetation pixels. A precision percentage of 93% in this model reflects a high proportion of positive instances or vegetation pixels correctly classified in relation to vegetation and non-vegetation pixels, although this proportion is lower compared to the two previous models. Similarly, regarding the recall metric, having a value of 93% indicates that the support vector machine model has a high capacity (though lower than KNN and decision trees) to identify vegetation pixels out of the total instances. As a consequence of the two previous metrics, the F1-score at a percentage value of 93.02% indicates a good balance for the classification of vegetation pixels, minimizing false positives and false negatives. Finally, it is worth mentioning that, unlike the two previous models, the precision obtained through 5-fold cross-validation is higher than the metrics derived from the confusion matrix (precision, recall, and F1-score). Likewise, a 95% precision value through cross-validation indicates high consistency in the model's performance with different datasets, even surpassing that achieved with decision trees in this case.

Continuing with the random forest model, which was implemented using the scikit-learn library with 200 estimators, an entropy criterion, and a depth of 5, the confusion matrix presented in Figure 8 was obtained. The confusion matrix obtained for this model indicates optimal performance in the

Table 3: Metrics obtained for the support vector machine model.

Metric	Results
Precision	0.93
Recall	0.93
F1-Score	0.9302
Accuracy obtained by cross-validation	0.95

Table 4: Metrics obtained for the random forest.

Metric	Results
Precision	1.0
Recall	1.0
F1-Score	1.0
Accuracy obtained by cross-validation	0.99

classification of vegetation pixels in hyperspectral images, similar to that obtained with the KNN model. In this regard, from the first row of the matrix, it is observed that 44 pixels were correctly classified as vegetation, constituting true positives. Similarly, in the second row, it can be seen that the 56 non-vegetation pixels were accurately identified by the model. Thus, the absence of false positives and false negatives in this matrix indicates an optimal level of precision in the random forest model.

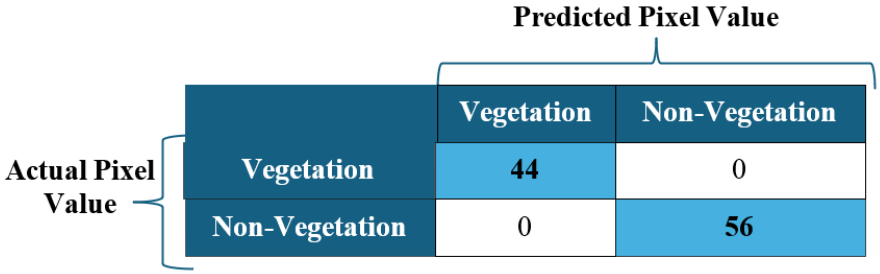


Figure 8: Confusion matrix for Random Forest.

Based on the results obtained in the confusion matrix presented in Figure 8, precision, recall, and F1-score metrics were derived, along with the precision metric obtained from the 5-fold cross-validation model. These results are detailed in Table 4.

Based on the results presented in Table 4, it is possible to observe how a perfect percentage is achieved in precision, recall, and F1-score metrics, similar to what was achieved with KNN. Thus, a percentage value of 100% in the precision metric indicates an optimal proportion of positive instances or vegetation pixels classified with respect to vegetation and non-vegetation pixels. Regarding the recall metric, it can be stated that having a percentage value of 100% indicates that the random forest model has an optimal capacity (similar to KNN) to identify positive instances (vegetation pixels) among the total positive or negative instances. As a consequence of the results obtained in the previous metrics, the percentage value obtained for F1-score of 100% indicates an optimal balance in the classification of positive instances or vegetation pixels, completely minimizing the presence of false positives and false negatives. Continuing with the logistic regression model, which was implemented using the scikit-learn library, the confusion matrix presented in Figure **??** was obtained. The confusion matrix obtained for logistic regression reveals an outstanding performance in the classification of vegetation pixels in hyperspectral images, being slightly surpassed by the KNN and random forest models. Thus, from the first row of the confusion matrix, it is observed that 44 vegetation pixels were correctly classified, constituting true positives. Similarly, in the second row, it is possible to observe how the 55 non-vegetation pixels have been correctly identified (true negatives), while 1 vegetation pixel was classified incorrectly (false negative). Thus, in the absence of false positives and the presence of only 1 false negative, the confusion matrix indicates an outstanding level of precision for the logistic regression model, being slightly surpassed by the KNN and random forest models.

Based on the results obtained from the confusion matrix presented in Figure 9, precision, recall, and

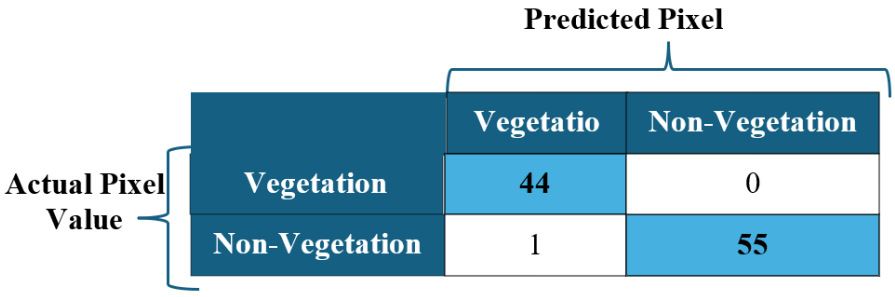


Figure 9: Confusion matrix for Logistic Regression.

Table 5: Metrics obtained for the logistic regression.

Metric	Results
Precision	0.99
Recall	0.99
F1-Score	0.99
Accuracy obtained by cross-validation	0.99

F1-score metrics were calculated, along with the precision metric obtained from the cross-validation model with 5 partitions (folds). These results are presented in detail in Table 5.

Based on the results presented in Table 5, it can be observed that a percentage of 99% is obtained in all four considered metrics: precision, recall, F1-score, and precision through cross-validation. A 99% precision metric indicates that a high proportion of positive instances or vegetation pixels are correctly classified with respect to positive and negative instances. Regarding the recall metric, it can be concluded that having a 99% value is indicative of the model having a very high capacity (second only to KNN and random forests) to identify positive instances (vegetation pixels) among the total positive or negative instances. From the results obtained in the two previous metrics, the achieved value for F1-score is 99%, representing a very high balance in the classification of positive instances or vegetation pixels, minimizing to a large extent the presence of false positives and false negatives. Finally, in order to make a comparison between the results obtained in the different models considered, Figure 10 presents a graph that allows comparing the precision metric through cross-validation and recall for the different models. The combination of precision through cross-validation and sensitivity (recall) as evaluation metrics to compare classification models in hyperspectral images offers a balanced perspective of performance. Precision reflects the general ability of the model to classify correctly with different validation sets, while sensitivity focuses on the ability to identify positives, which is crucial in cases where the omission of positive instances can have significant implications.

The comparison of five models for the classification of vegetation and non-vegetation pixels indicates variations in their performance. Among them, the KNN model stands out as the most accurate, achieving a precision of 99.3% through cross-validation, while achieving a sensitivity of 100%. This indicates an exceptional ability to correctly identify positive instances. On the other hand, Decision Trees (DT), Random Forest (RF), and Logistic Regression (LR) models exhibit high precisions of 98.6%, 99%, and 99%, respectively, in cross-validation, with sensitivity ranging from 98% to 100%. While the SVM model, although maintaining a respectable precision of 95%, shows a slightly lower sensitivity of 93%. Overall, these results suggest that, in this specific context of pixel classification in hyperspectral images, the KNN model stands out as the most robust and accurate.

In order to deploy the best model obtained, the detection of vegetation pixels was performed on the hyperspectral image presented in Figure 2, which has a total of 2,250,000 pixels and was used to obtain the sample of 400 pixels that formed the dataset. Thus, with the support of the spectral library in Python, the image was vectorized, and each pixel was iterated through one by one, being classified by the KNN model. The result is shown in Figure 11, where pixels painted in blue correspond to vegetation areas. From the model execution, it was found that the marked vegetation areas in the image accounted for a total of 293,048 pixels, corresponding to 13.042% of the total image area. The validity of the KNN model as a reference for vegetation detection is

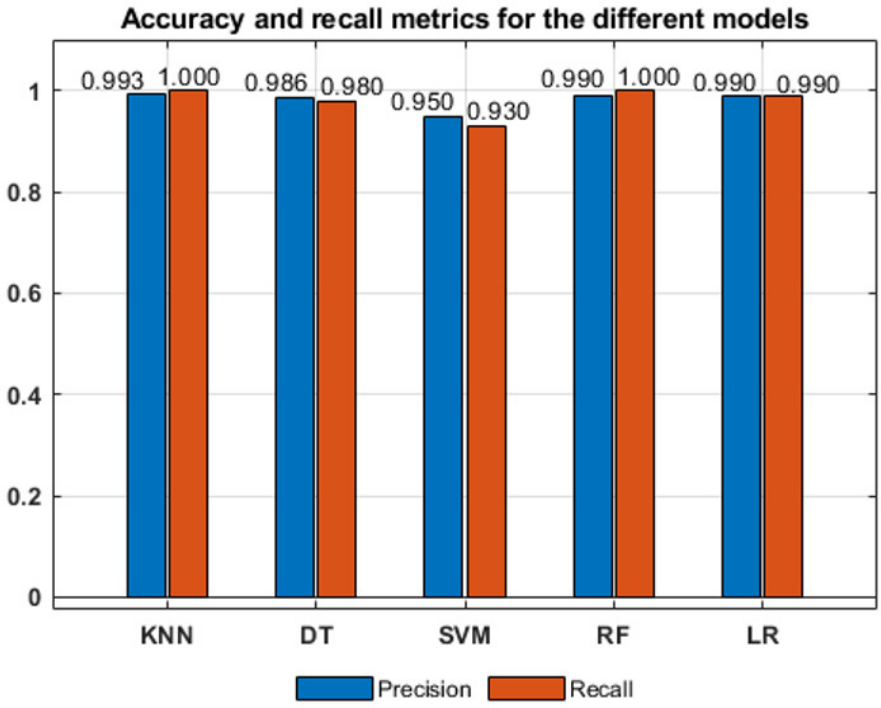


Figure 10: Accuracy and recall metrics for the different models.

supported by a meticulous visual inspection of the reference hyperspectral image. This inspection reveals a highly coherent correspondence between the areas identified by the KNN model and the actual vegetation areas. The accuracy of this correlation supports confidence in the model's ability to accurately discriminate vegetation. This analysis not only positions the model as an effective tool for vegetation detection but also suggests the possibility of extrapolation for the identification of other materials. Given the consistency observed in vegetation detection, it is plausible to assume that the KNN model, trained with the appropriate hyperspectral information, can be successfully applied to the identification of various materials. This extrapolation is based on the premise that the distinctive spectral characteristics enabling precise vegetation identification may also be relevant for detecting other elements. Additionally, the results emphasize the vital importance of hyperspectral images in monitoring vegetation over time in urban environments, establishing a crucial connection with carbon dioxide (CO2) levels. These images allow detailed monitoring of the evolution of vegetation cover, providing fundamental insights into the response of urban vegetation to factors such as urbanization, climate changes, and atmospheric pollution.

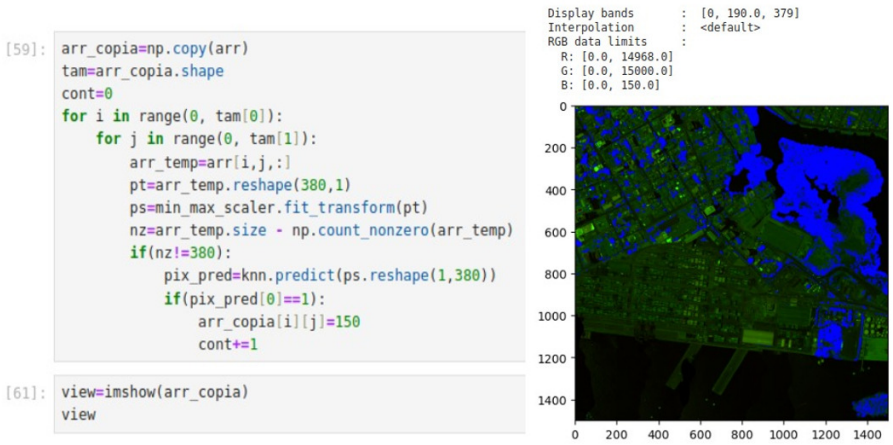


Figure 11: Vegetation detection in the reference hyperspectral image.

In the context of discussion, it is important to mention that the research presented in this article

introduces an approach based on artificial intelligence, specifically machine learning, for the identification of vegetation in hyperspectral images. Initially, a dataset of vegetation and non-vegetation pixels was formed, consisting of a total of 400 normalized pixels, each with a total of 380 spectral bands. From this dataset, supervised learning models (KNN, decision trees, support vector machines, random forests, and logistic regression) were evaluated, obtaining precision results through cross-validation between 95% and 99%. This indicates that these methods are highly effective in recognizing spectral signatures and can be applied for the detection of different types of materials. In this regard, the model that obtained the best performance was KNN with 3 neighbors, with cross-validated precision of 0.993 and a sensitivity or recall metric of 1.0. Thus, the results presented in this article are a relevant contribution in terms of the effectiveness of machine learning methods compared to the approach presented in [23], where the use of distance metrics for vegetation detection in hyperspectral images is proposed. However, it should be mentioned that, although supervised learning methods have demonstrated higher effectiveness in terms of percentage precision, they still face challenges in computational efficiency compared to the use of distance metrics. Therefore, future work aims to hybridize supervised learning methods with parallel computing methods to not only improve result effectiveness but also efficiency. In the same vein, concerning the work proposed in Torres-Gil et al. [11], where the detection of materials in hyperspectral images is proposed based on the use of relevant maxima and minima, which may vary depending on the purity of the material, this work proposes an effective method based on precision metrics in cross-validation with ratings exceeding 95%. However, as mentioned earlier, challenges in computational efficiency must be addressed in future work. Despite this, the work proposed in [11] provides an important idea to consider in the future context of machine learning, which is the identification of bands that show a higher correlation with the predictor attribute to consider a subset of these to improve computational efficiency.

Additionally, it is worth mentioning that one of the additional contributions of this work has been the use of open-source tools from the Python language for hyperspectral image processing and the application of supervised learning techniques. The spectral library, along with the scikit-learn library and its associated libraries (numpy, pandas, matplotlib), proved to be effective for developing the different phases considered in the CRISP-DM methodology. In this regard, this work is a fundamental contribution for research centers in developing countries, given the costs of proprietary licensed tools like ENVI, which has been used in various hyperspectral image analysis projects [68, 69]. However, the advantage of using open-source tools is not limited to licensing but also extends to flexibility in integrating and customizing material detection methods in hyperspectral images. This enables the integration of different artificial intelligence approaches, such as neural networks, unsupervised learning, fuzzy logic, or parallel computing, for the processing and analysis of hyperspectral images. Thus, this research paves the way for hybridization with other methods to improve, for example, computational performance when processing hyperspectral images.

4 Conclusions

In this article, we proposed an approach based on machine learning, specifically focusing on supervised learning models for vegetation detection in hyperspectral images. Through experi-mentation with a dataset containing 400 spectral signatures, each with 380 bands, it was found that all five supervised learning models considered in this article (KNN, decision trees, support vector machines, random forests, and logistic regression) achieved accuracy percentages above 95% through 5-fold cross-validation. This approach proved highly effective for detecting spectral sig-natures of vegetation compared to other methods such as distance and correlation. In this regard, the best-performing model in the comparison was KNN (see Figure 10 and Table 1), which achieved an accuracy percentage of 99.3% through cross-validation. Thus, the approach proposed in this article aims to be extrapolated to the detection of other types of materials in hyperspectral images, intending to validate its effectiveness in these contexts as well. For the comparison of the machine learning models fitted in this research, a dataset with 400 spectral signatures was employed (200 vegetation signatures and 200 non-vegetation signatures), each with 380 reflectance bands. This dataset was conformed from samples taken from a reference hyper-spectral image of the Manga neighborhood in the city of Cartagena de Indias, Colom-

bia, containing 2,250,000 pixels and 380 frequency bands. Although the samples seemingly correspond to a smaller percentage of the total pixels in the image, the amount of information included in a single pixel across its different bands and the distinctive characteristics of the spectral signatures associated with each material, particularly vegetation in this case, have allowed supervised learning models to effectively recognize vegetation in these images. In the same vein, and to facilitate the recognition of hyperspectral signatures by supervised learning models, the normalization of reflectance values for each of the 380 bands in the dataset is crucial. Considering that currently, for hyperspectral image processing and material detection in such images, widely disseminated tools with proprietary licenses like ENVI are prevalent, the results obtained in this article demonstrated the effectiveness of open-source tools and technologies in the development of various phases of the considered methodology. Thus, for the processing and access to information of different pixels in the images, the spectral Python library was utilized. For the vectorization of pixels in hyperspectral images, the numpy library in Python was employed. For accessing, storing, and reading the dataset in comma-separated files, the pandas library in Python was used. Finally, for the development of processes involving normalization, fitting, and evaluation of supervised learning models, the advantages provided by the scikit-learn library in Python were harnessed. It is noteworthy, however, that the advantage in using these tools is not only represented in terms of licensing costs for educational and research centers in developing countries but also in the potential to hybridize different computational methods to enhance the effectiveness and effi-ciency of vegetation detection. With regard to other approaches found in the literature for vegetation detection in hyperspectral images, such as those related to the use of distance metrics, the supervised learning methods compared in this research proved to be more effective in detecting vegetation in hyperspectral images, achieving accuracy rates exceeding 95% in all cases. Despite this, these models are computationally less efficient than those based on distance metrics. Therefore, it is necessary to propose hybrid computational methods that allow for the improvement of efficiency while main-taining the effectiveness demonstrated in this research.

Finally, as future work derived from this research, the following objectives are intended: a) assess the model's effectiveness with asbestos-cement, as this research is part of the project "Formulation of an integral strategy to reduce the impact on public and environmental health due to the presence of asbestos-cement in the territory of the Department of Bolivar" and serves as an initial step to evaluate methods for asbestos-cement detection in hyperspectral images, with the aim of extrapolating the proposed method to this domain; b) determine the reflectance bands that exhibit higher correlation with the predictor attribute, such that these bands are employed to adjust and validate the models to verify their effectiveness and enhance their efficiency; c) propose a hybrid approach that combines methods based on supervised learning and parallel computing to improve efficiency in vegetation detection in hyperspectral images.

Funding

This article is considered a product in the framework of the project "Formulation of an integral strategy to reduce the impact on public and environmental health due to the presence of asbestos in the territory of the Department of Bolivar", financed by the General System of Royalties of Colombia (SGR) and identified with the code BPIN 2020000100366. This project was executed by the University of Cartagena, Colombia, and the Asbestos-Free Colombia Foundation. Finally, the authors thank Federico Frassy for his support on the hyperspectral data management and classification, Aiken Hernando Ortega Heredia, María Angélica Narváez Cuadro, Carlos Andrés Castrillón Ortiz, Michelle Cecilia Montero Acosta, Margareth Peña Castro, Carlos David Arroyo Angulo and the rest of the research group team for logistical support. Finally, the authors thank Juan Manuel Gonzales of BlackSquare company for the support in the hyperspectral data acquisition.

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Pérez-Cabello, F.; Montorio, R.; and Alves, D. B.; (2021). Remote sensing techniques to assess post-fire vegetation recovery, *Curr. Opin. Environ. Sci. Heal.*, vol. 21, p. 100251, Jun. 2021, doi: 10.1016/J.COESH.2021.100251.

[2] Andreatta, D.; Gianelle, D.; Scotton, M.; and Dalponte, M.; (2022). Estimating grassland vegetation cover with remote sensing: A comparison between Landsat-8, Sentinel-2 and PlanetScope imagery, *Ecol. Indic.*, vol. 141, p. 109102, 2022, doi: 10.1016/j.ecolind.2022.109102.

[3] Sripada, R. P.; (2005). Determining In-Season Nitrogen Requirements for Corn Using Aerial Color-Infrared Photography, *North Carolina State University*.

[4] Shikwambana, L.; Xongo, K.; Mashalane, M.; and Mhangara, P.; (2023). Climatic and Vegetation Response Patterns over South Africa during the 2010/2011 and 2015/2016 Strong ENSO Phases, *Atmosphere (Basel)*, vol. 14, no. 2, pp. 1–14, 2023, doi: 10.3390/atmos14020416.

[5] García-Pardo, K. A.; Moreno-Rangel, D.; Domínguez-Amarillo, S.; and García-Chávez, J. R.; (2022). Remote sensing for the assessment of ecosystem services provided by urban vegetation: A review of the methods applied, *Urban For. Urban Green.*, vol. 74, p. 127636, Aug. 2022, doi: 10.1016/J.UFUG.2022.127636.

[6] Neinavaz, E.; Schlerf, M.; Darvishzadeh, R.; Gerhards, M.; and Skidmore, A. K.; (2021). Thermal infrared remote sensing of vegetation: Current status and perspectives, *Int. J. Appl. Earth Obs. Geoinf.*, vol. 102, p. 102415, Oct. 2021, doi: 10.1016/J.JAG.2021.102415.

[7] Meusburger, K.; Bänninger, D.; and Alewell, C.; (2010). Estimating vegetation parameter for soil erosion assessment in an alpine catchment by means of QuickBird imagery, *Int. J. Appl. Earth Obs. Geoinf.*, vol. 12, no. 3, pp. 201–207, Jun. 2010, doi: 10.1016/J.JAG.2010.02.009.

[8] Henrich, V.; Krauss, G.; Götze, C.; and Sandow, C.; Index DataBase. A database for remote sensing indices. 2012, https://www.indexdatabase.de/info/credits.php

[9] Huang, S.; Tang, L.; Hupy, J. P.; Wang, Y.; and Shao, G.; (2021). A commentary review on the use of normalized difference vegetation index (NDVI) in the era of popular remote sensing, *J. For. Res.*, vol. 32, no. 1, pp. 1–6, 2021, doi: 10.1007/s11676-020-01155-1.

[10] Beltrán Hernández, D. H.; (2017). Aplicación de índices de vegetación para evaluar procesos de restauración ecológica en el Parque Forestal Embalse del Neusa, *Universidad Militar Nueva Granada*, Neusa, Colombia, 2017.

[11] Torres Gil, L. K.; Valdelamar Martínez, D.; and Saba, M.; (2023). The Widespread Use of Remote Sensing in Asbestos, Vegetation, Oil and Gas, and Geology Applications, *Atmosphere (Basel)*, vol. 14, no. 1, p. 172, 2023, doi: 10.3390/atmos14010172.

[12] Kauth, R. J.; and Thomas, G. S. P.; (1976). The tasselled cap - A graphic description of the spectral-temporal development of agricultural crops as seen by Landsat, *Symposium of Machine Processing of Remotely-Sensed Data: Proc. of the LARS*, Purdue University, West Lafayette, Indiana, pp. 4B41–4B51, Jun. 1979.

[13] Tucker, C. J.; (1979). Red and photographic infrared linear combinations for monitoring vegetation, *Remote Sens. Environ.*, vol. 8, no. 2, pp. 127–150, May 1979, doi: 10.1016/0034-4257(79)90013-0.

[14] Huete, A. R.; (1988). A soil-adjusted vegetation index (SAVI), *Remote Sens. Environ.*, vol. 25, no. 3, pp. 295–309, Aug. 1988, doi: 10.1016/0034-4257(88)90106-X.

[15] Crippen, R. E.; (1990). Calculating the vegetation index faster, *Remote Sens. Environ.*, vol. 34, no. 1, pp. 71–73, Oct. 1990, doi: 10.1016/0034-4257(90)90085-Z.

[16] Gitelson, A. A.; and Merzlyak, M. N.; (1998). Remote sensing of chlorophyll concentration in higher plant leaves, *Adv. Sp. Res.*, vol. 22, no. 5, pp. 689–692, Jan. 1998, doi: 10.1016/S0273-1177(97)01133-2.

[17] Bannari, A.; Asalhi, H.; and Teillet, P. M.; (2002). Transformed difference vegetation index (TDVI) for vegetation cover mapping, *IEEE International Geoscience and Remote Sensing Symposium*, pp. 3053–3055 vol.5, 2002, doi: 10.1109/IGARSS.2002.1026867.

[18] MaxMax; Enhanced Normalized Difference Vegetation Index (ENDVI).

[19] Wolf, A. F.; (2012). Using WorldView-2 Vis-NIR multispectral imagery to support land mapping and feature extraction using normalized difference index ratios, *Proc. SPIE*, vol. 8390, pp. 188–195, May 2012, doi: 10.1117/12.917717.

[20] Shore, S. N.; (2003). Astrochemistry, *Encycl. Phys. Sci. Technol.*, pp. 665–678, 2003, doi: 10.1016/B0-12-227410-5/00032-6.

[21] Galle, N. J.; et al.; (2021). Correlation of WorldView-3 spectral vegetation indices and soil health indicators of individual urban trees with exceptions to topsoil disturbance, *City Environ. Interact.*, vol. 11, p. 100068, 2021, doi: 10.1016/j.cacint.2021.100068.

[22] Davies, B. F. R.; et al.; (2023). Multi- and hyperspectral classification of soft-bottom intertidal vegetation using a spectral library for coastal biodiversity remote sensing, *Remote Sens. Environ.*, vol. 290, p. 113554, 2023, doi: 10.1016/j.rse.2023.113554.

[23] Chanchí Golondrino, G. E.; Ospina Alarcón, M. A.; and Saba, M.; (2023). Vegetation Identification in Hyperspectral Images Using Distance/Correlation Metrics, *Atmosphere (Basel).*, vol. 14, no. 7, p. 1148, Jul. 2023, doi: 10.3390/atmos14071148.

[24] Smyth, T. A. G.; Wilson, R.; Rooney, P.; and Yates, K. L.; (2022). Extent, accuracy and repeatability of bare sand and vegetation cover in dunes mapped from aerial imagery is highly variable, *Aeolian Res.*, vol. 56, p. 100799, 2022, doi: 10.1016/j.aeolia.2022.100799.

[25] Tian, J.; et al.; (2023). Simultaneous estimation of fractional cover of photosynthetic and non-photosynthetic vegetation using visible-near infrared satellite imagery, *Remote Sens. Environ.*, vol. 290, p. 113549, 2023, doi: 10.1016/j.rse.2023.113549.

[26] Adão, T.; et al.; (2017). Hyperspectral Imaging: A Review on UAV-Based Sensors, Data Processing and Applications for Agriculture and Forestry, *Remote Sens.*, vol. 9, no. 11, p. 1110, Oct. 2017, doi: 10.3390/RS9111110.

[27] Wan, L.; Li, H.; Li, C.; Wang, A.; Yang, Y.; and Wang, P.; (2022). Hyperspectral Sensing of Plant Diseases: Principle and Methods, *Agronomy*, vol. 12, no. 6, pp. 1–19, 2022, doi: 10.3390/agronomy12061451.

[28] Wang, S.; et al.; (2023). Airborne hyperspectral imaging of cover crops through radiative transfer process-guided machine learning, *Remote Sens. Environ.*, vol. 285, p. 113386, 2023, doi: 10.1016/j.rse.2022.113386.

[29] Khan, A.; Vibhute, A. D.; Mali, S.; and Patil, C. H.; (2022). A systematic review on hyperspectral imaging technology with a machine and deep learning methodology for agricultural applications, *Ecol. Inform.*, vol. 69, p. 101678, 2022, doi: 10.1016/j.ecoinf.2022.101678.

[30] Chen, D.; et al.; (2022). Improved Na+ estimation from hyperspectral data of saline vegetation by machine learning, *Comput. Electron. Agric.*, vol. 196, p. 106862, 2022, doi: 10.1016/j.compag.2022.106862.

[31] Gakhar, S.; and Tiwari, K. C.; (2021). Spectral – spatial urban target detection for hyperspectral remote sensing data using artificial neural network, *Egypt. J. Remote Sens. Sp. Sci.*, vol. 24, no. 2, pp. 173–180, 2021, doi: 10.1016/j.ejrs.2021.01.002.

[32] Ma, B.; Zeng, W.; Hu, G.; Cao, R.; Cui, D.; and Zhang, T.; (2022). Normalized difference vegetation index prediction based on the delta downscaling method and back-propagation artificial neural network under climate change in the Sanjiangyuan region, China, *Ecol. Inform.*, vol. 72, p. 101883, 2022, doi: 10.1016/j.ecoinf.2022.101883.

[33] Trombetti, M.; Riaño, D.; Rubio, M. A.; Cheng, Y. B.; and Ustin, S. L.; (2008). Multitemporal vegetation canopy water content retrieval and interpretation using artificial neural networks for the continental USA, *Remote Sens. Environ.*, vol. 112, no. 1, pp. 203–215, 2008, doi: 10.1016/j.rse.2007.04.013.

[34] Badola, A.; Panda, S. K.; Roberts, D. A.; Waigl, C. F.; Jandt, R. R.; and Bhatt, U. S.; (2022). A novel method to simulate AVIRIS-NG hyperspectral image from Sentinel-2 image for improved vegetation/wildfire fuel mapping, boreal Alaska, *Int. J. Appl. Earth Obs. Geoinf.*, vol. 112, p. 102891, 2022, doi: 10.1016/j.jag.2022.102891.

[35] Rumpf, T.; Mahlein, A.-K.; Steiner, U.; Oerke, E.-C.; Dehne, H.-W.; and Plümer, L.; (2010). Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance, *Comput. Electron. Agric.*, vol. 74, no. 1, pp. 91–99, 2010, doi: 10.1016/j.compag.2010.06.009.

[36] Wang, L.; and Wang, Q.; (2022). Fast spatial-spectral random forests for thick cloud removal of hyperspectral images, *Int. J. Appl. Earth Obs. Geoinf.*, vol. 112, p. 102916, 2022, doi: 10.1016/j.jag.2022.102916.

[37] Ding, X.; Wang, Q.; and Tong, X.; (2022). Integrating 250 m MODIS data in spectral unmixing for 500 m fractional vegetation cover estimation, *Int. J. Appl. Earth Obs. Geoinf.*, vol. 111, p. 102860, 2022, doi: 10.1016/j.jag.2022.102860.

[38] Martínez-Plumed, F.; et al.; (2021). CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories, *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 8, pp. 3048–3061, Aug. 2021, doi: 10.1109/TKDE.2019.2962680.

[39] Yun, Z.; Weihua, L.; and Yang, C.; (2014). Applying balanced scorecard strategic performance management to CRISP-DM, *Proc. 2014 Int. Conf. Inf. Sci. Electron. Electr. Eng.*, pp. 2009–2014, Apr. 2014, doi: 10.1109/InfoSEEE.2014.6946275.

[40] Nock, R.; Sebban, M.; and Bernard, D.; (2003). A simple locally adaptive nearest neighbor rule with application to pollution forecasting, *Int. J. Pattern Recognit. Artif. Intell.*, vol. 17, no. 08, pp. 1369–1382, Dec. 2003, doi: 10.1142/S0218001403002952.

[41] Atiya, A. F.; (2005). Estimating the Posterior Probabilities Using the K -Nearest Neighbor Rule, *Neural Comput.*, vol. 17, no. 3, pp. 731–740, Mar. 2005, doi: 10.1162/0899766053019971.

[42] Toussaint, G.; (2005). Geometric proximity graphs for improving nearest neighbor methods in instance-based learning and data mining, *Int. J. Comput. Geom. Appl.*, vol. 15, no. 02, pp. 101–150, Apr. 2005, doi: 10.1142/S0218195905001622.

[43] Lee, H.; (2018). K-Nearest Neighbor rule using Distance Information Fusion, *J. Korean Inst. Intell. Syst.*, vol. 28, no. 2, pp. 160–163, Apr. 2018, doi: 10.5391/JKIIS.2018.28.2.160.

[44] Zhou, Z.; Li, Z.; Cai, Z.; and Wang, P.; (2019). Fault Identification Using Fast k-Nearest Neighbor Reconstruction, *Processes*, vol. 7, no. 6, p. 340, Jun. 2019, doi: 10.3390/pr7060340.

[45] Kang, S.; (2021). k-Nearest Neighbor Learning with Graph Neural Networks, *Mathematics*, vol. 9, no. 8, p. 830, Apr. 2021, doi: 10.3390/math9080830.

[46] Doru, C.; and Costel, B.; (2023). Classification of Image Classes Based on the PCA Algorithm Optimized by the KNN Algorithm Improved by Genetic Algorithms, *2023 6th International Conference on Information Communication and Signal Processing (ICICSP), IEEE*, pp. 103–108, Sep. 2023, doi: 10.1109/ICICSP59554.2023.10390821.

[47] Abe, S.; (2020). Minimal Complexity Support Vector Machines for Pattern Classification, *Computers*, vol. 9, no. 4, p. 88, Nov. 2020, doi: 10.3390/computers9040088.

[48] Harikiran, J. D. J. H.; (2020). Hyperspectral image classification using support vector machines, *IAES Int. J. Artif. Intell.*, vol. 9, no. 4, p. 684, Dec. 2020, doi: 10.11591/ijai.v9.i4.pp684-690.

[49] Díaz-Vico, D.; Prada, J.; Omari, A.; and Dorronsoro, J.; (2020). Deep support vector neural networks, *Integr. Comput. Aided. Eng.*, vol. 27, no. 4, pp. 389–402, Sep. 2020, doi: 10.3233/ICA-200635.

[50] Ioannou, C.; and Vassiliou, V.; (2021). Network Attack Classification in IoT Using Support Vector Machines, *J. Sens. Actuator Networks*, vol. 10, no. 3, p. 58, Aug. 2021, doi: 10.3390/jsan10030058.

[51] Huang, Y.; Yang, G.; Xu, Y.; and Zhou, H.; (2021). Differential Privacy Principal Component Analysis for Support Vector Machines, *Secur. Commun. Networks*, vol. 2021, pp. 1–12, Jul. 2021, doi: 10.1155/2021/5542283.

[52] Ravi, S. R.; (2021). Naturally Generated Decision Trees for Image Classification, *[Online]*. Available: DOI not provided.

[53] Xu, S.; Liu, S.; Wang, H.; Chen, W.; Zhang, F.; and Xiao, Z.; (2020). A Hyperspectral Image Classification Approach Based on Feature Fusion and Multi-Layered Gradient Boosting Decision Trees, *Entropy*, vol. 23, no. 1, p. 20, Dec. 2020, doi: 10.3390/e23010020.

[54] Li, F.; Zhang, X.; and Ji, Y.; (2023). Decision Tree Model to Classify Wastewater Evaporation, *Ind. Eng. Chem. Res.*, vol. 62, no. 20, pp. 8111–8117, May 2023, doi: 10.1021/acs.iecr.3c00250.

[55] Jeon, Y.; and Cho, H.; (2019). Model based hybrid decision tree, *J. Korean Data Inf. Sci. Soc.*, vol. 30, no. 3, pp. 515–524, May 2019, doi: 10.7465/jkdi.2019.30.3.515.

[56] Yao, J.; (2022). Carbon Sequestration Model Based on Decision Tree Regression and Logic Tree Model, *Acad. J. Environ. Earth Sci.*, vol. 4, no. 3, 2022, doi: 10.25236/AJEE.2022.040306.

[57] Jiang, Y.; (2019). Personalized Thermal Comfort Model with Decision Tree, *Intell. Control Autom.*, vol. 10, no. 4, pp. 168–177, 2019, doi: 10.4236/ica.2019.104012.

[58] Jin, B.; et al.; (2022). Determination of viability and vigor of naturally-aged rice seeds using hyperspectral imaging with machine learning, *Infrared Phys. Technol.*, vol. 122, p. 104097, 2022, doi: 10.1016/j.infrared.2022.104097.

[59] LaValley, M. P.; (2008). Logistic regression, *Circulation*, vol. 117, no. 18, pp. 2395–2399, 2008, doi: 10.1161/CIRCULATIONAHA.106.682658.

[60] Saha, D.; and Manickavasagan, A.; (2021). Machine learning techniques for analysis of hyperspectral images to determine quality of food products: A review, *Curr. Res. Food Sci.*, vol. 4, pp. 28–44, 2021, doi: 10.1016/j.crfs.2021.01.002.

[61] Lever, J.; Krzywinski, M.; and Altman, N.; (2016). Logistic regression, *Nat. Methods*, vol. 13, no. 7, pp. 541–542, Jul. 2016, doi: 10.1038/nmeth.3904.

[62] Zaidi, A.; and Al Luhayb, A. S. M.; (2023). Two Statistical Approaches to Justify the Use of the Logistic Function in Binary Logistic Regression, *Math. Probl. Eng.*, vol. 2023, no. 1, Jan. 2023, doi: 10.1155/2023/5525675.

[63] Wang, Z.; Hu, M.; and Zhai, G.; (2018). Application of deep learning architectures for accurate and rapid detection of internal mechanical damage of blueberry using hyperspectral transmittance data, *Sensors*, vol. 18, no. 4, p. 1126, 2018.

[64] Shebl, A.; Abriha, D.; Fahil, A. S.; El-Dokouny, H. A.; Elrasheed, A. A.; and Csámer, Á.; (2023). PRISMA hyperspectral data for lithological mapping in the Egyptian Eastern Desert: Evaluating the support vector machine, random forest, and XG boost machine learning algorithms, *Ore Geol. Rev.*, vol. 161, p. 105652, 2023, doi: 10.1016/j.oregeorev.2023.105652.

[65] Flynn, K. C.; Baath, G.; Lee, T. O.; Gowda, P.; and Northup, B.; (2023). Hyperspectral reflectance and machine learning to monitor legume biomass and nitrogen accumulation, *Comput. Electron. Agric.*, vol. 211, p. 107991, 2023, doi: 10.1016/j.compag.2023.107991.

[66] Yu, F.; Bai, J.; Jin, Z.; Gou, Z.; Yang, J.; and Chen, C.; (2023). Combining the critical nitrogen concentration and machine learning algorithms to estimate nitrogen deficiency in rice from UAV hyperspectral data, *J. Integr. Agric.*, vol. 22, no. 4, pp. 1216–1229, 2023, doi: 10.1016/j.jia.2022.12.007.

[67] Zambrano-Prado, P.; et al.; (2020). Laboratory-based spectral data acquisition of roof materials, *Int. J. Remote Sens.*, vol. 41, no. 23, pp. 9180–9205, Dec. 2020, doi: 10.1080/01431161.2020.1798548.

[68] Curcio, A. C.; Barbero, L.; and Peralta, G.; (2023). UAV-Hyperspectral Imaging to Estimate Species Distribution in Salt Marshes: A Case Study in the Cadiz Bay (SW Spain), *Remote Sens.*, vol. 15, no. 5, 2023, doi: 10.3390/rs15051419.

[69] ESRI; (2023). ENVI, 2023.

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Chanchí-Golondrino, G.; Ospina-Alarcón, M., Saba, M. (2025). Advancing hyperspectral image analysis: harnessing machine learning for precision vegetation identification in urban areas, *International Journal of Computers Communications & Control*, 20(5), 6923, 2025.
https://doi.org/10.15837/ijccc.2025.5.6923