INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 2, Month: April, Year: 2026
Article Number: 7229, https://doi.org/10.15837/ijccc.2026.2.7229

Detection of Vegetation Spectral Signatures in Hyperspectral Images using Artificial Neural Networks

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

Manuel Ospina-Alarcón*
Faculty of Engineering, Systems Engineering Program
University of Cartagena, Colombia
Cll. 30 N 39 B-192, Piedra de Bolívar, Ave. del Consulado, Cartagena-Colombia
*Corresponding author: mospinaa@unicartagena.edu.co

Gabriel Chanchí-Golondrino
Faculty of Engineering, Systems Engineering Program
University of Cartagena, Colombia
Cll. 30 N 39 B-192, Piedra de Bolívar, Ave. del Consulado, Cartagena-Colombia
gchanchig@unicartagena.edu.co

Manuel Saba
Faculty of Engineering, Civil Engineering Program
University of Cartagena, Colombia
Cll. 30 N 39 B-192, Piedra de Bolívar, Ave. del Consulado, Cartagena-Colombia
msaba@unicartagena.edu.co

Abstract

Considering the challenge in hyperspectral imaging of developing new computational methods that strike a balance between accurate material classification and computational complexity, this work proposes the design and tunability of a model based on a sequential artificial neural network (ANN) to classify vegetation in hyperspectral images with 380 bands. To carry out this research, an adaptation of the CRISP-DM methodology was used, structured into four phases: P1. Business and data understanding, P2. Data preparation, P3. Modeling and evaluation, and P4. Modl application. As a result, a sequential ANN model was developed, featuring 380 input layers and a single output layer, along with a set of dense layers containing 12, 8 and 4 artificial neurons. After 20 epochs, the model showed high performance and consistent behavior in the training and test sets under the experimental setup considered. The model was applied to a hyperspectral image of the Manga neighborhood in Cartagena, classifying 41.921% of the image pixels as vegetation. This percentage of points exceeds by 12.941% the percentage obtained by the spectral differential similarity method, in which less continuous point detections were observed. This method is a viable alternative for use in environmental monitoring systems, especially when applied in parallel to large-scale images.

Keywords: vegetation detection, hyperspectral imaging, environmental monitoring, artificial neural networks, remote sensing.

1 Introduction

The Remote sensing can be understood as a technique aimed at obtaining information about objects or areas on the Earth's surface without direct physical contact, using sensors mounted on platforms such as satellites or aircraft that capture the electromagnetic energy reflected or emitted by these objects [5], [16], [24]. In recent years, remote sensing has evolved significantly through its integration with unmanned aerial vehicles (UAVs) powered by global navigation satellite systems (GNSS), enabling its applicability and dissemination in various fields such as environmental monitoring and precision agriculture.

Unlike traditional methods that require point-by-point sampling of areas, remote sensing enables the observation of large or extensive areas in an immediate manner [2], [28], [32]. In the same sense, through remote sensing it is possible to conduct repetitive and systematic monitoring over time, which makes it possible to detect long-term changes and trends in the environment, soil, water, and vegetation [1], [43]. Based on the above, remote sensing allows for the saving of human and material resources by reducing the need for travel and the number of field tasks performed [15].

One of the most widespread remote sensing techniques is hyperspectral imaging, which is defined as an advanced imaging technology that combines spatial and spectral information, such that it can be considered as a "data cube" storing data in three dimensions: two spatial (x, y) and one spectral (λ), thus enabling the analysis of the composition and properties of materials for each pixel in the image [7], [8], [17], [34]. In this way, since hyperspectral images are composed of hundreds of narrow and continuous spectral bands, each pixel contains a complete spectrum, which allows for material identification, chemical composition analysis, and the retrieval of structural and functional details of the sample [29]. Unlike multispectral images, which capture data in a limited number of bands (typically between 3 and 10), hyperspectral images consist of dozens or hundreds of narrow and contiguous spectral bands, which implies a higher spectral resolution, allowing the distinction of materials that appear similar in multispectral images [26], [44], [47].

There are various studies in which the relevance of using spectral images in the environmental context can be appreciated. Thus, in [3], [4], [13], [40], hyperspectral images have been effectively used for the identification and quantification of microplastics in seawater and agricultural soils, in such a way that algorithms such as support vector machines (SVM) and convolutional neural networks (CNN) have shown high accuracy in classifying different types of polymers. In [41], [42], low-cost hyperspectral systems were developed for identifying spectral variations in minerals and soils, enabling improvements in the spatial resolution of environmental analyses, as well as better access to and use of remote sensing techniques. In [46], hyperspectral images were used together with radar data for vegetation detection in wetlands, through the application of ensemble methods by stacking models such as Random Forest, CatBoost, and XGBoost, thus achieving competitive alternative performance compared to these same models individually. In [35], multispectral images from Madrid (Spain) and certain areas of Indonesia were used for vegetation change detection through the Spectral Angle Mapper (SAM) method combined with support vector machines (SVM), resulting in high detection accuracy. In [6], [27], [36], conventional methods such as NDVI and classic machine learning techniques (random forest, support vector machines (SVM), and decision trees) were used for vegetation detection in high-resolution aerial images (RGB and NIR), multispectral images, and LiDAR data. Finally, in [11], distance and correlation techniques were used for vegetation detection in hyperspectral images with 380 bands from the city of Cartagena.

At the level of the existing challenges in the field of hyperspectral imaging, it is important to highlight, first of all, that the high dimensionality of spectral datacubes generates spectral and spatial redundancy, which increases complexity and computational cost, thus affecting the efficiency of classification algorithms [10], [18], [21]. In the same sense, in remote sensing application scenarios such as environmental monitoring (detection of pollutants, microplastics, or plant species), where it is necessary to identify and differentiate materials in large-scale images, it is essential to have algorithms with good computational efficiency that enable the processing of these large data volumes within reasonable timeframes and with limited resources [30], [31], [45]. On the other hand, although advanced models such as convolutional neural networks (CNNs) offer good performance in detection tasks, they are computationally and memory intensive, making it necessary to balance accuracy and efficiency

through hybrid techniques and dimensionality reduction methods [19], [22].

Based on the aforementioned challenges regarding the high dimensionality of hyperspectral images and considering that artificial neural networks (ANNs), due to their simple architectures and few layers, require fewer computational and memory resources than CNNs [14], [25], this work proposes, as its contribution, the evaluation of a method based on a sequential ANN architecture for the detection of spectral signatures of vegetation in hyperspectral images of the city of Cartagena, Colombia. While recurrent neural networks have been widely used to model sequential and temporal dependencies in spectral data [20], [33], their computational cost and architectural complexity may limit their suitability for large-scale pixel-wise processing in hyperspectral images. In this work, a feed-forward artificial neural network (ANN) of the multilayer perceptron (MLP) type is employed, implemented using the Sequential API provided by the TensorFlow/Keras framework. Although the term "sequential" refers here to the way layers are stacked and executed in order, the proposed model does not incorporate recurrent connections, hidden states, or temporal memory mechanisms. Instead, each pixel is processed independently based solely on its spectral signature across the 380 reflectance bands. This architectural choice allows for a simpler model with lower computational and memory requirements compared to recurrent or convolutional architectures, while still preserving strong discriminative capability for vegetation detection in hyperspectral images.

The proposed method was compared with the spectral differential similarity method in terms of classified vegetation area, spatial continuity of the resulting maps, and computational efficiency, which has been widely used for material detection in spectral images [12]. For the implementation of the proposed method, the advantages offered by open-source libraries and technologies such as Spectral, NumPy, Pandas, Matplotlib, and TensorFlow were used, aiming to make this work applicable for research centers and universities in developing countries for experimentation with hyperspectral images, considering the high cost of proprietary processing tools. In the same sense, based on the results obtained, this method aims to be integrated into real-time environmental monitoring systems, for example, for the detection of vegetation and water bodies.

It is important to emphasize that this work does not aim to propose a novel artificial neural network architecture. Instead, its main contribution lies in the applied and comparative evaluation of a simple multilayer perceptron (MLP) model for vegetation detection in hyperspectral images under a realistic operational scenario. While numerous studies have explored advanced neural architectures for hyperspectral image classification, this work focuses on analyzing the behavior, limitations, and computational trade-offs of a lightweight MLP model when compared with a classical spectral-based method. This positioning allows the proposed approach to be interpreted as a practical baseline and reference for applications where model simplicity, transparency, and computational cost are critical factors.

The remainder of the article is organized as follows: Section 2 presents the way in which the phases of the CRISP-DM methodology were adapted for the development of this research. Section 3 presents the results obtained through this study, which include: the creation of the dataset with spectral signatures of vegetation and non-vegetation; the structuring of the sequential artificial neural network (ANN); the tuning and evaluation of the model using the accuracy metric; the trained model was applied on a reference hyperspectral image; and the comparison of the results with the spectral differential similarity method. Finally, Section 4 presents the conclusions and future work derived from this research.

2 Methodology

For the development of this work, the CRISP-DM data mining and science methodology was adapted into four phases: P1. Business and data understanding, P2. Data preparation, P3. Modeling and evaluation, and P4. Model application (see Figure 1). The CRISP-DM methodology was chosen considering that it is robust, features well-defined and described phases associated with the lifecycle of a data science project, and facilitates planning, customization, documentation, and communication within the work team, thus enabling the process to be efficient and repeatable [9], [37], [38].

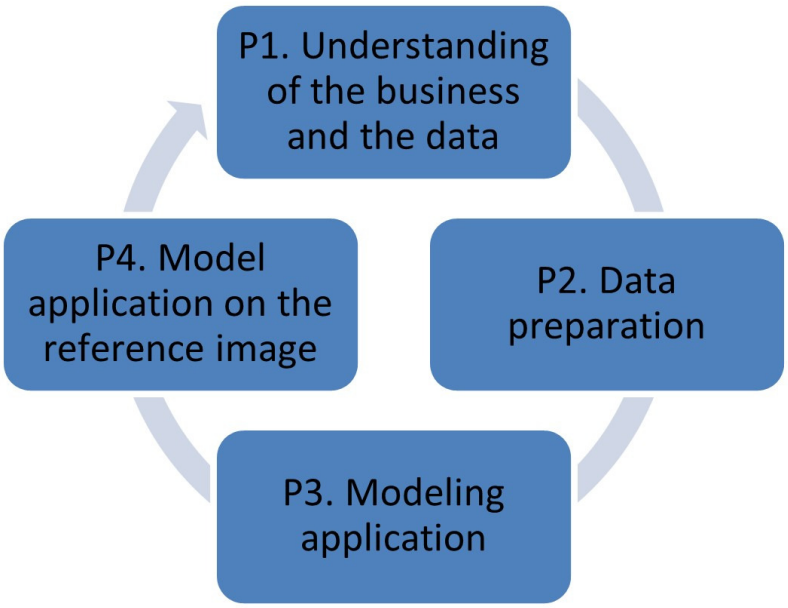

Figure 1: Methodology considered

In Phase 1 of the methodology, a total of 2000 sample spectral signatures were selected (1000 from vegetation and 1000 from other materials (roofs, water bodies, roads, among others)) from a reference hyperspectral image of the Manga neighborhood in the city of Cartagena, which has a total of 725x850 pixels and 380 reflectance bands. It is worth noting that the considered hyperspectral image is composed of 380 spectral bands, distributed over the wavelength range between 400 and 2400 nm, with an approximate spectral sampling interval of 5 nm, which includes bands corresponding to the visible (VIS), near-infrared (NIR), and shortwave infrared (SWIR) regions.

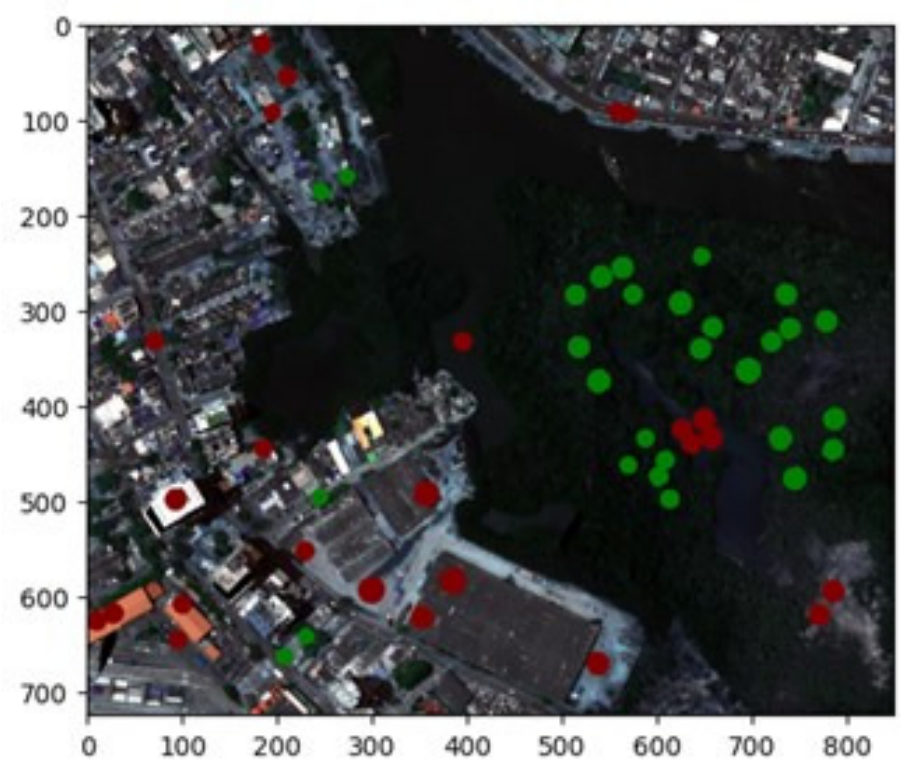

Figure 2: Data acquisition: Selected sample pixels

The acquisition of the sample pixels or spectral signatures was carried out through visual inspection using the ImageJ image analysis tool, such that once these pixels were captured, their correspondence

with vegetation pixels was verified using the spectral signature. These spectral signatures allow for the differentiation of various materials and were used to create a training dataset. In Figure 2, the selected vegetation pixels or signatures are shown in green on an RGB representation of the reference hyperspectral image, while the selected pixels or sample signatures of other materials are shown in red. It is worth additionally noting that the circular regions shown in Figure 2 include multiple points and are defined according to the sampling criterion employed in the ImageJ tool.

In this regard, Figure 3 shows the spectral signatures of the two groups of sample pixels, in such a way that for each sample pixel, the value of normalized reflectance across the 380 spectral bands can be observed. It is possible to see from Figure 3 how the spectral signatures of vegetation exhibit a set of relevant peaks between band 100 and band 300, which enable their differentiation from the signatures of other materials. It is worth noting that the spectral signatures are normalized within the range from 0 to 1, since a normalization process was previously applied to the entire image by identifying the minimum and maximum reflectance values of the original image. This procedure was carried out using the functions provided by the NumPy library, which allowed the spectral values to be uniformly scaled prior to the extraction and analysis of the signatures

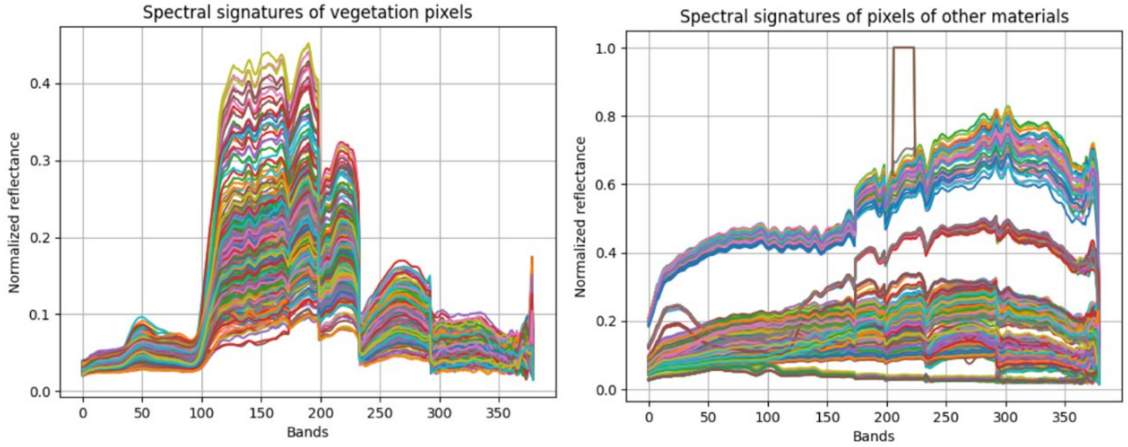

Figure 3: Data acquisition: Spectral signatures of vegetation and other materials samples

In phase 2 of the methodology, a dataset was created with the 2000 normalized spectral signatures presented in Figure 3. In this way, the 1000 vegetation spectral signatures were labeled with a 1, while the 1000 spectral signatures of other materials were labeled with a 0. The integration of the 2000 signatures was performed using the vstack() function from the Python numpy library, such that the unified array was converted into a pandas dataframe structure. This type of structure is widely used for dataset representation in Python and for efficient data processing in machine learning processes.

In phase 3 of the methodology, the architecture of the sequential ANN was structured using the TensorFlow library, considering 380 inputs, a set of hidden layers, and one output. In this way, after each hidden layer, a ReLU activation function was used, and for the output layer, a sigmoid activation function was applied. Additionally, the model was trained with the training set and evaluated using the test set over 100 epochs. The model architecture was defined using the Sequential class from TensorFlow, incorporating regularization through Dropout in each hidden layer to prevent overfitting. Furthermore, the Adam optimizer was employed with a learning rate of 0.001, and the binary_crossentropy loss function was used due to the binary nature of the problem. At the end of each training cycle, the model's performance was evaluated using the test set, with accuracy being the primary evaluation metric. Specifically, the implementation of the architecture using TensorFlow can be mathematically modeled through Eq. 1.

$$\hat{y} = f\left(\sum_{i=1}^{n} \omega_i x_i + b\right) \tag{1}$$

Where x_i represents the inputs of the neuron, ω_i the weights associated with each input, b is the bias, $f(\cdot)$ is the activation function (such as ReLU, tanh, sigmoid), and $\hat{y}$ is the estimated output of

the neuron

Finally, in Phase 4 of the methodology, the application of the trained feed-forward ANN model to the reference hyperspectral image of the Manga neighborhood in the city of Cartagena was performed. In this way, the prediction of the possible classification categories for each pixel of the reference hyperspectral image was made, and the vegetation pixels were colored green in an RGB representation of the image. Additionally, at the time of deploying the model on the image, the number of vegetation pixels was counted and, consequently, the percentage of vegetation pixels relative to the total image was calculated. This percentage of vegetation pixels was compared with the percentage of pixels classified by the spectral differential similarity (SDS) method, which was deployed similarly on the reference image after identifying the detection thresholds. It is worth noting that the SDS method is computed from the normalized summation of the differences between the 380 bands of the characteristic vegetation spectral signature (X_i) and the bands of the spectral signature of a given image pixel (Y_i), such that when the normalized summation approaches zero, the evaluated pixel corresponds to a vegetation pixel. Based on this, a similarity measure derived from the normalized summation of differences (see Eq. 2) was defined and subsequently calibrated by identifying the detection thresholds using vegetation sample pixels and sample pixels corresponding to other materials.

$$\text{SDS} = 100 - \left(\frac{\sum |X_i - Y_i|}{380}\right) \times 100 \tag{2}$$

Additionally, the proposed method's efficiency was compared to the spectral differential similarity method by running both on a 50x50 pixel image with 380 bands, with repetitions of 25, 50, 75, and 100. This aimed to obtain the average time per repetition for each method and the total average time each method takes to process the image region. Based on the preceding analysis, the relative efficiency of the proposed method was determined by comparing it to the spectral differential similarity method, one of the approaches with the lowest computational cost.

3 Results and Discussion

Regarding the obtained results, first, a dataset was created from a reference hyperspectral image with 380 reflectance bands, consisting of 1000 vegetation spectral signatures labeled with 1 and 1000 spectral signatures of other materials labeled with 0, forming a pandas dataframe structure as shown in Table 1. From Table 1, it is possible to observe how the vegetation pixels or signatures were labeled with a 1, while the pixels of other materials were labeled with a 0. Likewise, it can be seen how the dataframe columns correspond to each of the 380 reflectance bands of each pixel. This dataset was used for model training with a training set (70%) and for evaluation using a test set (30%).

Table 1: Dataset for model fitting

band8	**band9**	**band10**	**...**	**band372**	**band373**	**band374**	**...**	**band380**	**pixel**
0.026592	0.028102	0.028825	...	0.051937	0.054301	0.056664	...	0.015102	1
0.031188	0.032173	0.032699	...	0.057715	0.059816	0.061917	...	0.015102	1
0.031845	0.033027	0.034012	...	0.045043	0.047209	0.048457	...	0.015102	1
...	...	...	...	...	...	...	...	...	...
0.051937	0.054301	0.056664	...	0.051937	0.054301	0.056664	...	0.015102	0
0.057715	0.059816	0.061917	...	0.057715	0.059816	0.061917	...	0.015102	0
0.045043	0.047209	0.048457	...	0.045043	0.047209	0.048457	...	0.015102	0

Now, for the structure of the sequential ANN model, different layer configurations were tested, identifying that the optimal model configuration consists of 380 inputs (due to the 380 bands of the dataset and the image), 3 hidden layers (12, 8, and 4 neurons), and one output layer, considering the binary nature of the problem. After each of the hidden layers, the ReLU activation function was used, except for the last layer where a sigmoid function was applied. The report with the description of the model output by TensorFlow and its architecture are presented in Figure 4.

```
model = Sequential()
model.add(Dense(units=12,activation='relu',input_shape=(380,),kernel_regularizer=regularizers.l2(1e-4)))
model.add(Dropout(0.3))
model.add(Dense(units=8,activation='relu',kernel_regularizer=regularizers.l2(1e-4)))
model.add(Dropout(0.3))
model.add(Dense(units=4,activation='relu',kernel_regularizer=regularizers.l2(1e-4)))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer=Adam(learning_rate=0.001),loss='binary_crossentropy',metrics=[
        'accuracy',
        Precision(name='precision',thresholds=0.5),
        Recall(name='recall',thresholds=0.5),
        F1Score(name='f1', threshold=0.5),
        AUC(name='auc')
    ])
model.summary()
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 12)	4,572
dropout (Dropout)	(None, 12)	0
dense_1 (Dense)	(None, 8)	104
dropout_1 (Dropout)	(None, 8)	0
dense_2 (Dense)	(None, 4)	36
dense_3 (Dense)	(None, 1)	5

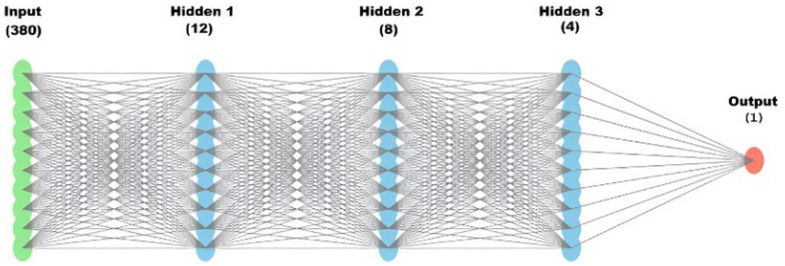

Figure 4: Architecture of the ANN model

Complementary to the information presented in Figure 4, Table 2 details the most relevant parameters used for the implementation, training, and evaluation of the ANN model.

Table 2: ANN model implementation, training, and evaluation parameters

Parameter	**Configuration**
Train/test split	70% / 30%
Global random seed	42 (Python, NumPy, and TensorFlow)
Computational determinism	Enabled (tf.config.experimental.enable_op_determinism)
Data scaling	Not specified in this fragment
ANN architecture	Dense layers with 12–8–4 neurons and sigmoid output
Regularization	L2 ($\lambda = 1\times10^{-4}$)
Dropout	0.3 (two layers)
Optimizer	Adam
Learning rate	0.001
Loss function	Binary cross-entropy
Batch size	32
Maximum epochs	100
Evaluation metrics	Accuracy, Precision, Recall, F1-score, AUC
GPU usage	Not used
Execution environment	Google Colab Pro L4
TensorFlow version	2.19

After defining the architecture of the ANN, the model was trained and validated over a total of 100 epochs with a batch size of 32, using the training and test sets, which were partitioned maintaining the 70% and 30% ratio. Based on the above, Figure 5 presents the evaluation of the model's performance in terms of loss and accuracy across the different considered epochs.

The graphs presented in Figure 5 reveal that both loss and accuracy exhibit a marked improvement throughout the training process. In the loss graph, the training loss decreases very rapidly during the first epochs, reaching low values within approximately the first 10–15 epochs and remaining relatively stable thereafter, with only minor fluctuations. The validation loss follows a similar decreasing trend, showing a sharp initial reduction and maintaining low values across most of the training, although slight oscillations are observed at later epochs. Overall, the gap between training and validation losses remains small for most of the epochs, suggesting a stable learning behavior without evident overfitting. On the other hand, in the accuracy graph, the model demonstrates a rapid increase in training accuracy during the initial epochs, reaching values close to 1.0 shortly after the beginning of training. The validation accuracy closely mirrors this behavior, remaining consistently high and stable

throughout the epochs. This strong alignment between training and validation accuracy indicates a stable learning behavior and consistent performance between the training and test sets under the considered experimental setup.

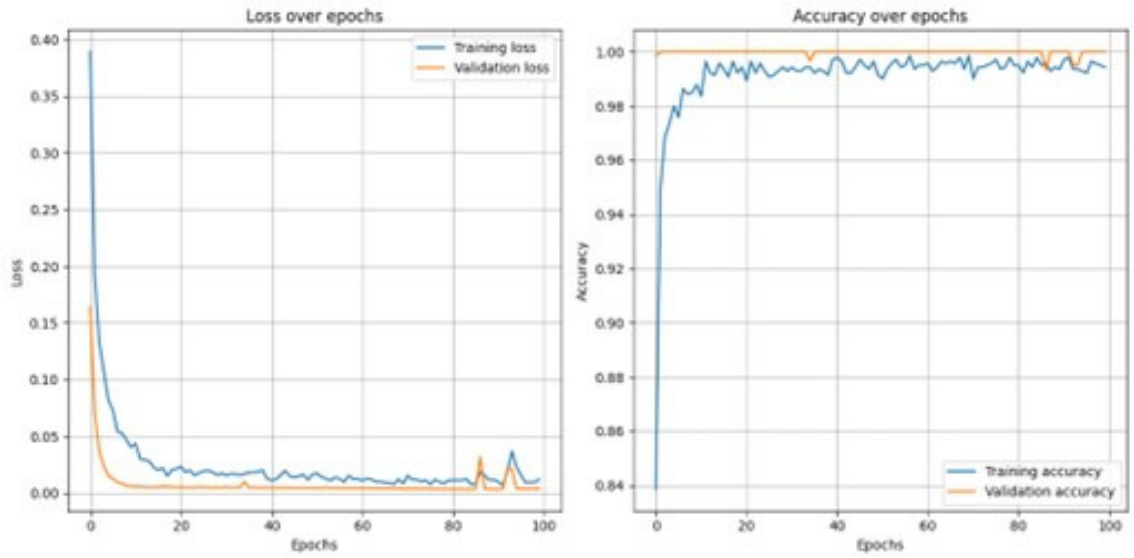

Figure 5: Evaluation of the model's performance and loss per epoch

The high values obtained for accuracy, precision, recall, and F1-score should be interpreted with caution, as they may suggest a risk of overfitting given the limited size of the labeled dataset. In this study, the dataset was randomly partitioned into training and test sets using a 70%–30% split, ensuring that the evaluation metrics were computed exclusively on samples not seen during training. No overlap was allowed between training and test pixels.

Regarding regularization, the simplicity of the multilayer perceptron architecture, together with the limited number of hidden neurons and the use of ReLU activations, was intentionally chosen to reduce model capacity and mitigate overfitting. In addition, the convergence behavior observed after approximately 30 epochs, where training and validation curves stabilize with a reduced gap, suggests that the model does not continue to memorize the training data beyond this point. Nevertheless, due to the reduced number of labeled samples, the reported performance values should be understood as indicative of model fitting under the considered experimental conditions rather than as evidence of universal generalization.

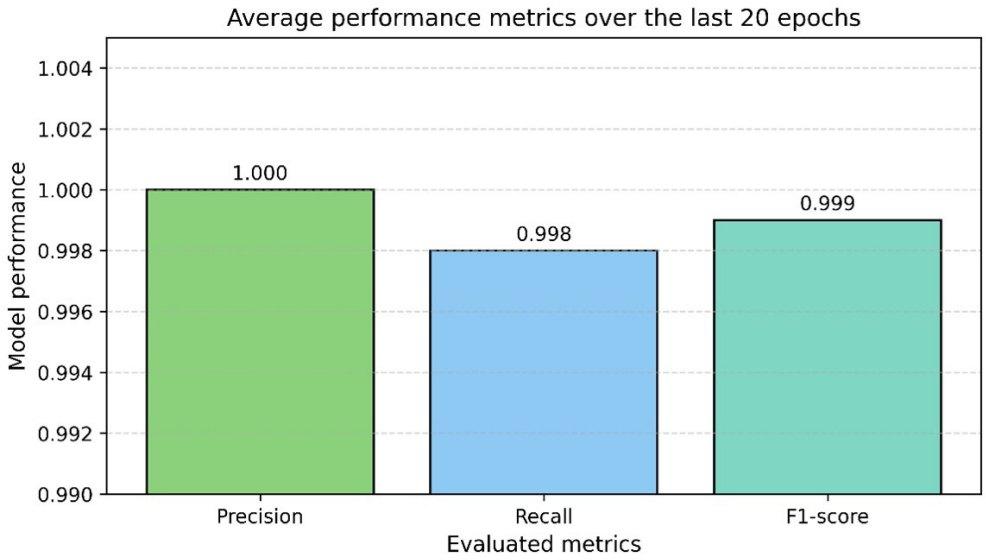

Figure 6: Average performance metrics over the last 20 epochs

Additionally, the average performance of the ANN model over the last 20 epochs was computed in terms of the precision, recall, and F1-score metrics, resulting in the graph presented in Figure 6.

From the results shown in Figure 6, it is possible to appreciate that the ANN model achieves a very high and stable performance in detecting vegetation pixels (1) against other materials (0), with average precision (1.000), recall (0.998), and F1-score (0.999) computed over the last 20 epochs. These results indicate an effective capability to discriminate both classes, with minimal presence of false positives and false negatives. In this sense, the combination of metrics close to unity evidences a robust balance between accuracy and coverage, while the averaging over the final stages of training suggests an adequate generalization behavior, associated with stable model convergence and the correct capture of spectral patterns representative of vegetation.

Once the model was trained and evaluated, identifying that it presents excellent fitting capability after 20 epochs, the trained ANN model was applied to the reference hyperspectral image to classify each pixel based on its spectral signature. In this way, the pixels predicted by the model with label 1 were painted dark green on an RGB representation of the reference image (see Figure 7).

In Figure 7, it is possible to observe how, initially, the hyperspectral image is flattened to match the format of the array that the model receives for prediction. Subsequently, data standardization is performed by leveraging the properties provided by the scaler previously fitted using the training data (70%), such that a new version of the input data to the ANN model is obtained with zero mean and unit standard deviation. After this step, prediction is executed, yielding binary labels. Following the prediction process, the labels identified as 1, which correspond to vegetation areas, are filtered and colored green in the RGB image. Finally, the total number of pixels classified as vegetation is counted, and the percentage relative to the total number of pixels in the image is calculated.

Additionally, it is possible to observe from Figure 7 how the ANN model classified 41.921% of the image pixels as vegetation on the reference hyperspectral image. In order to compare the behavior of the proposed method with respect to the spectral differential similarity (SDS) method, the SDS approach was also implemented and evaluated using the same reference image and the same set of sample pixels.

It is important to note that the comparison between the proposed ANN-based method and the SDS approach is not intended to provide a pixel-wise accuracy assessment, as no exhaustive ground truth labeling is available for the reference hyperspectral image. Instead, the comparison focuses on the relative proportion of pixels classified as vegetation and on the spatial continuity of the resulting vegetation maps when both methods are applied under identical conditions.

```
imagen = cv2.imread(ruta+"/cartagena_test.png")
img_copy = imagen.copy()
#Flatten the normalized image (arr_norm) to 2D
arr_flat = arr_norm.reshape(-1, arr_norm.shape[-1])
#Standardize the array (arr_flat) using a previously fitted scaler
arr_scaled = scaler.transform(arr_flat)
#Make predictions with the model
y_prob = model.predict(arr_scaled)
#Convert probabilities to binary labels
y_pred = (y_prob > 0.5).astype(int)
#Ensure y_pred has the same dimensions as the original image
y_pred_reshaped = y_pred.reshape(arr_norm.shape[0], arr_norm.shape[1])
#Modify the RGB image using the predictions
#Assign blue color (0, 0, 255) where the prediction is 1
img_copy[y_pred_reshaped == 1] = [0, 128, 0]
#Counting the number of pixels
cont_pix = np.count_nonzero(y_pred_reshaped == 1)
por_pix=(cont_pix/(tam[0]*tam[1]))*100
print("Perc of asb", round(por_pix,3),"%")
```

```
19258/19258 ━━━━━━━━━━━━━━━━━━━━ 24s 1ms/step
Perc of asb 41.921 %
```

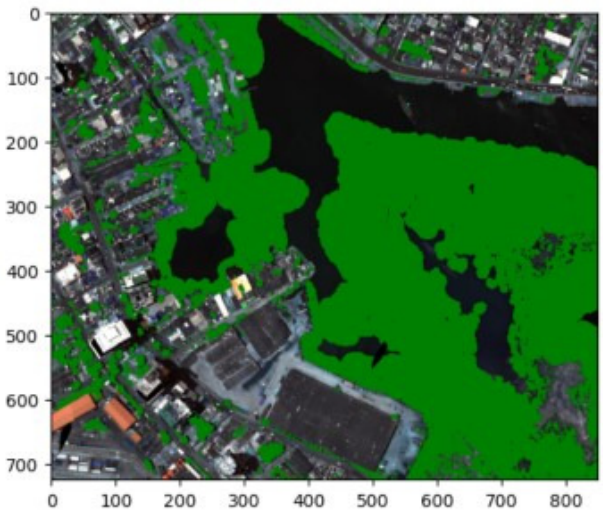

Figure 7: Application of the trained ANN model to the reference hyperspectral image

Figure 8 presents the evaluation of the SDS method using the 1000 vegetation spectral signatures and the 1000 spectral signatures of other materials, where the minimum similarity obtained for vegetation pixels and the maximum similarity obtained for other materials can be observed. It is worth

noting that these percentages were obtained by determining the minimum and maximum values after applying Eq. 2 to the vegetation and non-vegetation sample pixels, respectively, in order to identify the detection threshold.

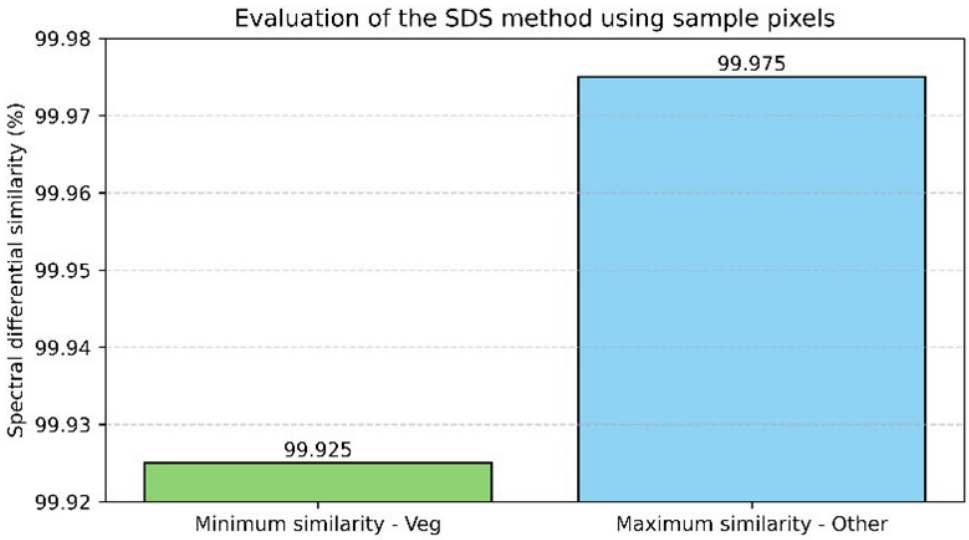

Figure 8: Evaluation of the SDS method using sample pixels

From Figure 8, it is possible to observe that with the SDS method, the maximum similarity obtained with pixels of other materials exceeds the minimum similarity obtained with vegetation pixels by 0.05%, such that there is overlap, and when counting the number of pixels of other materials that exceed the 99.925% threshold, it is found that the method confuses 39.8% of the signatures of other materials with vegetation signatures. Based on this, and in order to avoid confusion in detection, a vegetation detection threshold of 99.975% was selected. This threshold, used for the training of the SDS method on the entire reference image, is presented in Figure 9, where the classified vegetation areas are shown on an RGB representation of the image dataset.

```
arr_copia=np.copy(arr)
img_copy = imagen.copy()
tam=arr_copia.shape
cont_pix=0
for i in range(tam[0]):
  for j in range(tam[1]):
    #Obtaining the pixel i,j of the image
    pix=arr_norm[i][j]
    #Obtaining the spectral difference
    diff=np.abs(pix-pix_prom)
    #Obtaining the percentage of similarity
    porc=100-(np.sum(diff)/380)
    if(porc>=99.975):
      img_copy[i][j]=(0,128,0)
      cont_pix+=1
      mat_dif[i][j]=1
porc_pix_veg=(cont_pix/(tam[0]*tam[1]))*100
print("Percentage of Veg:", round(porc_pix_veg,3),"%")
```

```
Percentage of Veg: 28.98 %
```

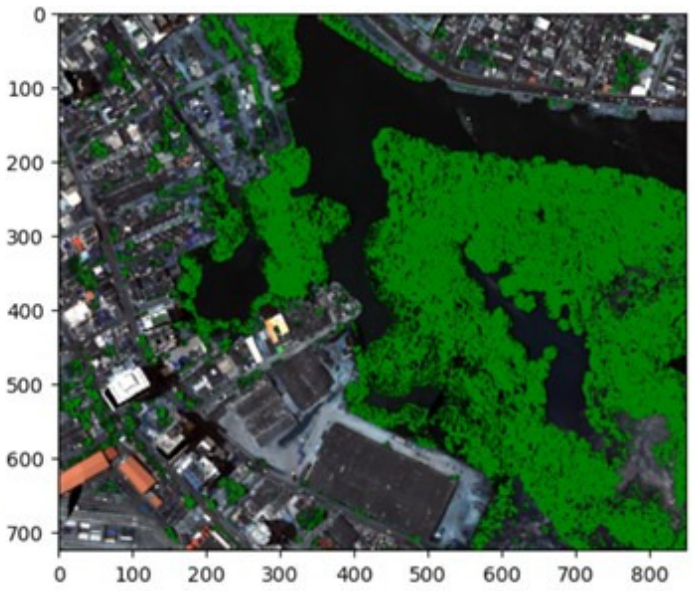

Figure 9: Application of the SDS method to the reference hyperspectral image

It can be observed from Figure 9 how the SDS method iterates over each pixel of the image, obtaining for each pixel a 380-dimensional spectral vector that is processed using the spectral differential similarity with a characteristic vegetation pixel. This reference pixel is obtained by averaging the 100

vegetation spectral signatures band by band. In this implementation, the SDS method was developed using the Python NumPy library. If the similarity percentage exceeds a threshold of 99.975%, the corresponding pixel is colored green in the RGB representation and counted as vegetation.

Using this procedure, the SDS method classified 28.98% of the pixels in the reference image as vegetation, which is 12.941% lower than the percentage obtained by the ANN-based method. In addition, the resulting vegetation map presents lower spatial continuity compared to the ANN-based classification, with visible gaps in areas that remain unclassified.

It is worth noting that the percentages of 41.921% and 28.98% correspond to the proportion of pixels detected as vegetation by the ANN and SDS methods, respectively, when applied to the entire hyperspectral image after calibration using the available sample signature dataset. In the case of the ANN model, 70% of the labeled signatures were used for training, whereas for the SDS method, which is not a machine learning approach, the full set of vegetation signatures was used to compute the detection threshold.

Beyond the comparison with the spectral differential similarity (SDS) method, it is relevant to contextualize the obtained results with respect to other widely used approaches for hyperspectral image classification. Methods such as support vector machines (SVM) and Random Forest have been extensively applied in hyperspectral vegetation detection due to their robustness when working with high-dimensional data and limited training samples. In many cases, these methods achieve strong classification performance but require careful parameter tuning and may exhibit increased computational cost when applied pixel-wise to large hyperspectral scenes.

Similarly, convolutional neural networks (CNNs) have demonstrated high detection capability by exploiting both spectral and spatial information; however, their deployment typically involves deeper architectures, higher memory consumption, and longer training and inference times. In contrast, the objective of this work is not to outperform all existing methods, but to analyze the behavior of a lightweight multilayer perceptron (MLP)-based ANN as a practical baseline under data-scarce and resource-constrained conditions. From this perspective, the proposed model offers an alternative with different computational and modeling trade-offs, emphasizing simplicity, transparency, and ease of implementation rather than maximum achievable accuracy.

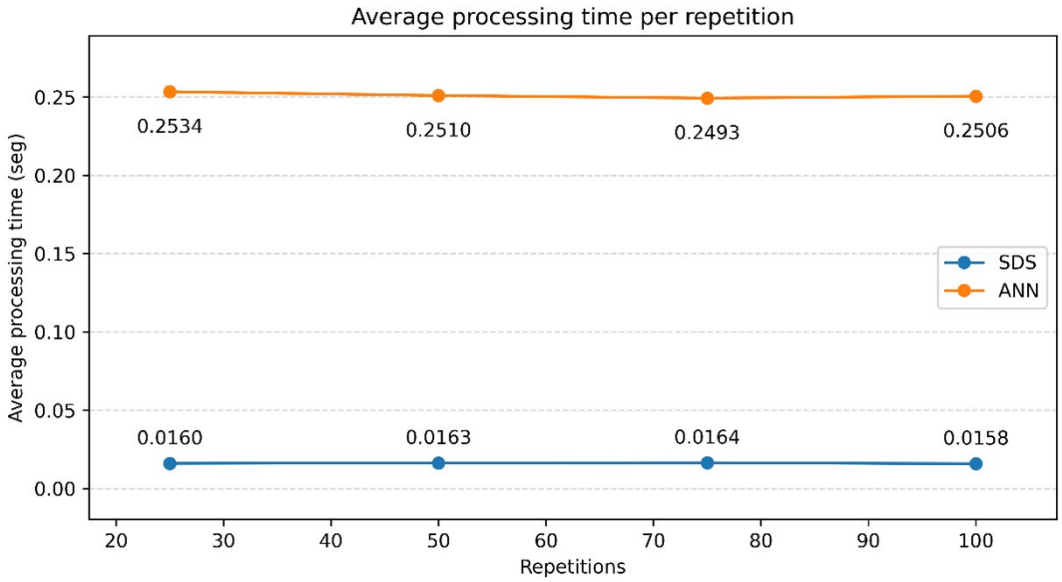

Figure 10: Average processing time per repetition

Now, in order to evaluate the computational efficiency of the proposed method compared to the SDS method, multiple repetitions of the methods (25, 50, 75, and 100) were performed on a 50x50 pixel region with 380 reflectance bands to identify the average time per repetition and the total average time spent by each method in processing the image region. It is worth noting that these experiments were conducted in the conventional academic environment of Google Colab Pro L4, with 53 GB of

RAM and 112.6 GB of available storage; under its standard configuration and without the use of GPU acceleration, these resources proved sufficient for the proper execution of the experiment, ensuring the stability of the runtime environment and the reproducibility of the obtained results. For the implementation of the multiple repetitions of the methods, the advantages provided by the timeit library were used, which allows for multiple executions of a routine or method and provides the time spent on the multiple executions. In this way, Figure 10 presents the times obtained by each of the two methods evaluated in the 4 groups of repetitions performed

From Figure 10, it is possible to observe that the SDS method achieves, on average, lower processing times than the ANN model when processing the same image region, with average times below 0.02 seconds across the different repetitions. On the other hand, for the ANN model, the average processing time for the different repetitions is around 0.25 seconds. In this way, the average time obtained for the SDS method across the different repetitions was 0.016 seconds, whereas the average time obtained for the ANN model was 0.251 seconds. In this way, it is possible to identify that the ANN model is 15.687 times less efficient than the SDS method.

From a methodological perspective, the contribution of this work should be interpreted as an applied and comparative study rather than as a proposal of a novel classification model. The results presented provide insight into how a simple MLP-based ANN behaves when applied to hyperspectral vegetation detection and how its outputs differ from those of a traditional spectral differential similarity approach. By explicitly analyzing classification behavior, spatial continuity of detected vegetation, and computational efficiency, this study contributes to a clearer understanding of the practical trade-offs involved in selecting lightweight neural models for hyperspectral image analysis.

As part of this discussion, the proposed ANN-based approach demonstrated a higher proportion of pixels classified as vegetation and improved spatial continuity when compared with the SDS method under the considered experimental setup. Specifically, the ANN classified 41.921% of the pixels as vegetation, whereas the SDS method classified 28.98%, resulting in a difference of 12.941% in the detected vegetation area, which is visually reflected in a more continuous vegetation mapping in the RGB representation of the hyperspectral image. These results suggest that the proposed lightweight MLP model is capable of capturing discriminative spectral patterns relevant for vegetation detection in hyperspectral data. Within this context, and considering the experimental conditions of this study, the obtained results are consistent with previous research based on distance and correlation methods [11], while offering an alternative with different computational and modeling trade-offs.

On the other hand, by using open-source libraries and technologies for image pre-processing as well as for model implementation, tuning, and evaluation, this work illustrates a fully reproducible workflow for hyperspectral image analysis in academic contexts. Given the high cost and limited flexibility of proprietary tools, the proposed approach provides an accessible alternative for institutions in developing countries to experiment with hyperspectral data. Moreover, studies that rely on proprietary platforms such as ENVI for material detection [23], [39] may benefit from integrating or extending their workflows with open-source components such as those used in this research. Thus, in this research, the spectral library was used for accessing the spectral band data of the image; the TensorFlow library was used for the implementation, tuning, and evaluation of the proposed model; the numpy library was used for the normalization of the reflectance of the original image and for the implementation of the SDS method; the pandas library was used to load the points with the coordinates of the two sample pixel groups; and matplotlib was used to generate the various statistical graphs obtained in this research.

It is important to note that the experimental evaluation presented in this study was conducted using a single reference hyperspectral image. As a result, the reported performance and comparative results should be interpreted within the context of this specific image and acquisition conditions. While the training and testing samples were extracted from different spatial locations within the image, allowing the model to learn and be evaluated on spectrally diverse pixels, the use of a single scene inherently limits the ability to fully assess generalization across different sensors, acquisition times, or environmental conditions.

Nevertheless, this experimental setup is representative of realistic operational scenarios in which limited labeled data are available, a common situation in hyperspectral remote sensing applications.

In this sense, the objective of the evaluation is not to claim universal generalization, but rather to analyze the relative behavior, stability, and computational trade-offs of a lightweight ANN model compared to a classical spectral-based method under identical conditions.

Regarding the limitations of the proposed method, it is important to note that although it has a less complex architecture than CNN-based models, its computational efficiency was lower than that of the SDS method, being approximately 13.63 times slower. While this difference may not be critical for small-scale images, it becomes relevant for large-area environmental monitoring scenarios, where processing time is a key factor. Consequently, future work should focus on optimizing the implementation through parallel computing or hardware acceleration strategies.

4 Conclusions

Considering the need for computational methods that provide a good balance between material classification in hyperspectral images and computational complexity, this work proposed as a contribution the evaluation of the fitting and prediction capability of a model supported by a sequential ANN architecture, which demonstrated, based on the accuracy metric, high precision and consistency in the training and test sets of a dataset extracted from a hyperspectral image with 380 reflectance bands. This study was conceived as an applied and comparative analysis, aiming to evaluate the suitability of a simple ANN model for hyperspectral vegetation detection rather than to introduce a new learning architecture. Based on this, the model aims to be considered as a reference for integration into environmental monitoring systems for the analysis of the temporal evolution of vegetation in the urban context. Furthermore, this model can be extrapolated and hybridized in the academic and business context for the detection of different types of materials in the domain of hyperspectral images.

The proposed method was evaluated through a comparative analysis with the SDS method, being deployed on a reference hyperspectral image of the Manga neighborhood in the city of Cartagena, allowing for the detection of vegetation areas in a more continuous manner than the SDS method. In this regard, the proposed method achieved a vegetation percentage of 41.921% in the Manga neighborhood image, surpassing the SDS method by 12.941%. The architecture that enabled these results for the proposed method consists of 380 inputs and one output, with hidden layers of 12, 8, 4 neurons in the middle, such that after the input and each hidden layer, ReLU functions were applied, while the last layer used a sigmoid activation function. Thus, this obtained architecture can be replicated and adapted for hyperspectral images with equal or smaller dimensions for the detection of vegetation or other materials.

In terms of computational efficiency, the proposed method was compared to the SDS method by performing multiple repetitions (25, 50, 75, and 100) on a region of the hyperspectral image with 50x50 pixels and 380 reflectance bands, obtaining that on average, the proposed method takes 0.230 seconds to process the image region, while the SDS method takes 0.017 seconds to process the same region on average. Thus, although the times obtained by the ANN model are acceptable, this model is 13.63 times less efficient than the SDS model. In this regard, considering the results obtained under the considered experimental setup, it is advisable for processing large coverage images, such as those obtained in environmental monitoring, to hybridize with parallel computing architectures.

Additionally, this work demonstrated the relevance and practical usefulness of open-source software tools and technologies for image pre-processing, as well as for model tuning, evaluation, and application. Thus, this work and the tools considered can be taken as a reference to replicate or extrapolate experimentation in material detection in hyperspectral images, especially by universities and research centers in developing countries, considering the high cost of proprietary tools for processing and analyzing these images. Furthermore, about proprietary tools, the ability to hybridize and customize different computational approaches provided by open-source software and free software stands out. Although high performance metrics were obtained, these results should be interpreted in light of the limited labeled dataset and the controlled experimental setup. The conclusions drawn in this work are therefore bounded by the characteristics of the analyzed hyperspectral image and should be interpreted as indicative of the model behavior under the considered experimental conditions.

Finally, as future work derived from this research, the following is intended: a) to identify the

architecture with the best fit for water body detection in hyperspectral images, and b) to adapt the proposed model into an architecture supported by parallel computing, making use, for example, of the advantages provided by the Python Dask framework.

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

Author contributions

Gabriel Chanchí-Golondrino (Conceptualization, Data curation, Formal analysis, Investigation, Project administration, Resources, supervision, validation, visualization, Writing – original draft, Writing – review and editing). Manuel Saba (Funding acquisition, Investigation, Project administration, Resources and validation). Manuel Ospina-Alarcón (Conceptualization, Formal analysis, Investigation, Methodology, Resources, supervision, Software, validation, visualization, Writing – original draft, Writing – review and editing).

Conflict of interest

The authors declare that there is no conflict of interest that could compromise the validity of the results presented

References

[1] Abd El-Ghany, N. M.; Abd El-Aziz, S. E.; Marei, S. S. (2020) A review: application of remote sensing as a promising strategy for insect pests and diseases management, Environmental Science and Pollution Research, 27(27), 33503–33515, 2020.

[2] Abdulraheem, M. I.; Zhang, W.; Li, S.; Moshayedi, A. J.; Farooque, A. A.; Hu, J. (2023) Advancement of remote sensing for soil measurements and applications: A comprehensive review, Sustainability, 15(21), 15444–15444, 2023.

[3] Ai, W.; Lyu, X.; Ersan, M. S.; Xiao, F. (2022) Application of hyperspectral imaging technology in the rapid identification of microplastics in farmland soil, Science of the Total Environment, 807, 151030–151030, 2022.

[4] Ali, M. A.; Lyu, X.; Ersan, M. S.; Xiao, F. (2024) Critical evaluation of hyperspectral imaging technology for detection and quantification of microplastics in soil, Journal of Hazardous Materials, 476, 135041–135041, 2024.

[5] Awange, J.; Kiema, J. (2019) Fundamentals of remote sensing, Springer, 115–123, 2019.

[6] Ayhan, B.; Kwan, C.; Budavari, B.; Lu, Y.; Perez, D.; Li, J. (2020) Vegetation detection using deep learning and conventional methods, Remote Sensing, 12(15), 2502–2502, 2020.

[7] Bodkin, A.; Sheinis, A.; Norton, A.; Daly, J.; Beaven, S.; Weinheimer, J. (2009) Snapshot hyperspectral imaging: the hyperpixel array camera, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV, 7334, 73340H–73340H, 2009.

[8] Burger, J.; Gowen, A. (2011) Data handling in hyperspectral image analysis, Chemometrics and Intelligent Laboratory Systems, 108(1), 13–22, 2011.

[9] Cazacu, M.; Titan, E. (2020) Adapting CRISP-DM for social sciences, BRAIN. Broad Research in Artificial Intelligence and Neuroscience, 11(2), 99–106, 2020.

[10] Chang, Y.-L.; Li, J.; Wang, S.; Chen, C.-Y. (2022) Consolidated convolutional neural network for hyperspectral image classification, Remote Sensing, 14(7), 1571–1571, 2022.

[11] Chanchí Golondrino, G. E.; Ospina Alarcón, M. A.; Saba, M. (2023) Vegetation identification in hyperspectral images using distance/correlation metrics, Atmosphere, 14(7), 1148–1148, 2023.

[12] Chanchí Golondrino, G. E.; Saba, M.; Ospina Alarcón, M. A. (2025) Propuesta de un método computacional para la detección de asbesto en imágenes hiperespectrales a partir de la similitud diferencial espectral, Revista Colombiana de Tecnologías de Avanzada, 1(45), 195–203, 2025.

[13] Chen, H. (2020) Microscopic hyperspectral image analysis via deep learning, Griffith University, 1–180, 2020.

[14] Desai, V.; Savani, V.; Patel, R. (2021) Plant classification based on leaves using artificial neural network, International Journal of Integrated Engineering, 13(6), 1–10, 2021.

[15] Fang, Q. (2024) The advantages of using remote sensing technology to monitor forest fires, Applied Computational Engineering, 60(1), 42–48, 2024.

[16] Fu, W.; Ma, J.; Chen, P.; Chen, F. (2020) Remote sensing satellites for digital earth, In Manual of Digital Earth, 55–123, 2020.

[17] Gao, L.; Smith, R. T. (2015) Optical hyperspectral imaging in microscopy and spectroscopy, Journal of Biophotonics, 8(6), 441–456, 2015.

[18] Ghanbari Azar, S.; Meshgini, S.; Yousefi Rezaii, T.; Beheshti, S. (2020) Hyperspectral image classification based on sparse modeling of spectral blocks, Neurocomputing, 407, 12–23, 2020.

[19] Ghous, U.; Sarfraz, M. S.; Ahmad, M.; Li, C.; Hong, D. (2024) EXNet: (2+1)D extreme xception net for hyperspectral image classification, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 17, 5159–5172, 2024.

[20] Haidarh, M.; Mu, C.; Liu, Y.; He, X. (2025) Exploring traditional, deep learning and hybrid methods for hyperspectral image classification: A review, Journal of Information Intelligence, In Press, 1–25, 2025.

[21] Hu, Q.; Wang, X.; Jiang, J.; Zhang, X.-P.; Ma, J. (2024) Exploring the spectral prior for hyperspectral image super-resolution, IEEE Transactions on Image Processing, 33, 5260–5272, 2024.

[22] Huang, H.; Tao, S. (2024) Hyperspectral image classification with token fusion on GPU, Computer Vision and Image Understanding, 249, 104198–104198, 2024.

[23] Hussein, S. J.; Merzah, Z. F. (2020) Analysis of hyperspectral remote sensing images for extraction geological rock types maps by geospatial techniques, IOP Conference Series: Materials Science and Engineering, 901(1), 012016–012016, 2020.

[24] Jiménez-López, A. F.; Jiménez-López, M.; Jiménez-López, F. R. (2015) Multispectral analysis of vegetation for remote sensing applications, ITECKNE, 12(2), 156–167, 2015.

[25] Kareem, S.; Hamad, Z. J.; Askar, S. (2021) An evaluation of CNN and ANN in prediction weather forecasting: A review, Sustainable Engineering and Innovation, 3(2), 148–159, 2021.

[26] Khan, M. J.; Khan, H. S.; Yousaf, A.; Khurshid, K.; Abbas, A. (2018) Modern trends in hyperspectral image analysis: A review, IEEE Access, 6, 14118–14129, 2018.

[27] Kurdi, F. T. (2021) Random forest machine learning technique for automatic vegetation detection and modelling in LiDAR data, International Journal of Environmental Sciences and Natural Resources, 28(2), 1–10, 2021.

[28] Lechner, A. M.; Foody, G. M.; Boyd, D. S. (2020) Applications in remote sensing to forest ecology and management, One Earth, 2(5), 405–412, 2020.

[29] Liu, G.; Wang, Y.; Zhang, Y.; Chen, Z.; Li, X. (2021) Combination of structured illumination microscopy with hyperspectral imaging for cell analysis, Analytical Chemistry, 93(29), 10056–10064, 2021.

[30] Loughlin, C.; Pieper, M.; Manolakis, D.; Bostick, R.; Weisner, A.; Cooley, T. (2020) Efficient hyperspectral target detection and identification with large spectral libraries, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, 6019–6028, 2020.

[31] Manolakis, D.; Truslow, E.; Pieper, M.; Cooley, T.; Brueggeman, M. (2014) Detection algorithms in hyperspectral imaging systems, IEEE Signal Processing Magazine, 31(1), 24–33, 2014.

[32] Navalgund, R.; Jayaraman, V.; Roy, P. S. (2007) Remote sensing applications: an overview, Current Science, 93(12), 1447–1466, 2007.

[33] Nargesi, M. H.; Amiriparian, J.; Bagherpour, H.; Kheiralipour, K. (2024) Detection of different adulteration in cinnamon powder using hyperspectral imaging and artificial neural network method, Results in Chemistry, 9, 101644–101644, 2024.

[34] Park, B. (2016) Future trends in hyperspectral imaging, NIR News, 27(1), 35–38, 2016.

[35] Renza, D.; Martinez, E.; Molina, I.; Ballesteros, D. M. (2017) Unsupervised change detection in a particular vegetation land cover type using spectral angle mapper, Advances in Space Research, 59(8), 2019–2031, 2017.

[36] Rougier, S.; Puissant, A.; Stumpf, A.; Lachiche, N. (2016) Comparison of sampling strategies for object-based classification of urban vegetation from very high resolution satellite images, International Journal of Applied Earth Observation and Geoinformation, 51, 60–73, 2016.

[37] Saltz, J. S. (2021) CRISP-DM for data science: strengths, weaknesses and potential next steps, IEEE International Conference on Big Data, 2337–2344, 2021.

[38] Schröer, C.; Kruse, F.; Gómez, J. M. (2021) A systematic literature review on applying CRISP-DM process model, Procedia Computer Science, 181, 526–534, 2021.

[39] Sekhon, A. S.; Singh, P.; Kaur, M.; Gill, J. S. (2024) Hyperspectral imaging of foodborne pathogens at colony and cellular levels for rapid identification in dairy products, Food Science and Nutrition, 12(1), 239–254, 2024.

[40] Shan, J.; Zhao, J.; Zhang, Y.; Liu, L.; Wu, F.; Wang, X. (2019) Simple and rapid detection of microplastics in seawater using hyperspectral imaging technology, Analytica Chimica Acta, 1050, 161–168, 2019.

[41] Stuart, M. B.; McGonigle, A. J. S.; Willmott, J. R. (2019) Hyperspectral imaging in environmental monitoring, Sensors, 19(14), 3071–3071, 2019.

[42] Stuart, M. B.; Davies, M.; Hobbs, M. J.; Pering, T. D.; McGonigle, A. J. S.; Willmott, J. R. (2022) High-resolution hyperspectral imaging using low-cost components, Sensors, 22(12), 4652–4652, 2022.

[43] Sun, Y.; Wang, D.; Li, L.; Ning, R.; Yu, S.; Gao, N. (2024) Application of remote sensing technology in water quality monitoring, Water Research, 267, 122546–122546, 2024.

[44] Verhoeven, G. (2018) Multispectral and hyperspectral imaging, Encyclopedia of Archaeological Sciences, 1–4, 2018.

[45] Yan, H.-F.; Zhao, Y.-Q.; Chan, J. C.-W.; Kong, S. G. (2024) Rapid hyperspectral anomaly detection using discriminative band selection, IEEE Transactions on Geoscience and Remote Sensing, 62, 1–18, 2024.

[46] Yao, H.; Zhang, X.; Li, Y.; Wang, Z.; Chen, J. (2022) Combination of hyperspectral and quad-polarization SAR images to classify marsh vegetation, Remote Sensing, 14(21), 5478–5478, 2022.

[47] Zahiri, Z.; Laefer, D. F.; Kurz, T.; Buckley, S.; Gowen, A. (2022) A comparison of ground-based hyperspectral imaging and red-edge multispectral imaging for façade material classification, Automation in Construction, 136, 104164–104164, 2022.

Cite this paper as:

Ospina-Alarcón, M.A.; Chanchí-Golondrino, G.E.; Saba, M. (2026). Detection of Vegetation Spectral Signatures in Hyperspectral Images using Artificial Neural Networks, *International Journal of Computers Communications & Control*, 21(2), 7229, 2026.
https://doi.org/10.15837/ijccc.2026.2.7229