Multisource Remote Sensing Data Visualization Using Machine Learning

Ioana Cristina Plajer, Alexandra Băicoianu, Luciana Majercsik, and Mihai Ivanovici, *Member, IEEE*

***Abstract*— With the availability of several remotely sensed data sources, the problem of efficiently visualizing the information from multisource data for improved Earth observation becomes an intriguing and challenging subject. Multispectral (MS) and hyperspectral (HS) images encompass a wealth of spectral data that standard RGB monitors cannot replicate directly. Thus, it is important to elaborate methods for accurately representing this information on conventional displays. These images, with tens to hundreds of spectral bands, contain relevant data about specific wavelengths that RGB channels cannot capture. Traditional visualization methods often use only a limited amount of the available spectral information, resulting in a significant loss of information. However, recent advances in artificial intelligence models have provided superior visualization techniques. These artificial Intelligence (AI)-based methods allow for more realistic and visually appealing representations, which are important for the information interpretation and direct identification of areas of interest. The main goal of our study is to process aggregated datasets from various sources using a fully connected neural network (FCNN), while considering visualization as a secondary objective. Given that our data come from a variety of sources, a significant emphasis in our study was placed on the preprocessing stage. In order to achieve a consistent visualization across datasets from different sources, proper preprocessing by standardization or normalization procedures is essential. Our research comprises numerous experiments to demonstrate the effectiveness of the proposed technique for image visualization.**

***Index Terms*— Multisource, multispectral (MS) and hyperspectral (HS) images, neural network, normalization, remote sensing, standardization, visualization.**

I. Introduction

THE unique spectral properties of different types of materials determine variations in the way they absorb light, thus giving each material a distinct spectral fingerprint. Multispectral (MS) and hyperspectral (HS) sensors are capable of capturing a rich spectral information that can cover hundreds of spectral bands, allowing them to detect even the smallest changes in the reflectance or radiance of objects. The resulting images produced by MS and HS sensors include additional spectral information about the chemical composition of objects. This additional information makes them extremely useful in a series of applications in areas, such as agriculture [1], [2], forestry [3], [4], environmental monitoring and ecology [5], [6], object detection [7], land cover classification [8], [9], as well as military and industrial fields.

In this article, machine learning techniques allow for notable progress in a large range of applications of remote sensing. Examples include a Siamese Transformer Network designed for HS image target detection [10] and some techniques for HS image denoising and anomaly detection [11], [12], [13], [14]. These advances highlight the growing ability of machine learning strategies to refine the understanding and interpretation of remote sensing data.

The first step in MS and HS interpretation is the visualization of the data in a comprehensive manner for human users. MS and HS images contain more spectral information than standard RGB channels can display, providing important data about specific wavelengths, beyond the RGB capabilities.

A realistically displayed spectral satellite image enables direct human interpretation and identification of areas of interest, but accurate visualization is challenging because of the need to compress information from numerous bands into three while preserving essential spatial and spectral detail.

Several visualization techniques have been proposed in the scientific literature in order to generate realistic RGB images from spectral images. These include band selection [15], [16], independent component analysis (ICA) [17], and principal component analysis (PCA) [18], [19] based methods, linear and nonlinear methods [20], and, relatively recent, machine learning approaches [21], [22].

While the classical method based on band selection uses a very small part of the available information, disregarding the information available in the other unused bands, better results of visualization with artificial intelligence (AI) models have already been reported in recent research papers [23]. Such methods, like the proposed one, allow for a more realistic rendering of colors and, at the same time, producing more appealing images (vivid colors, good contrast, and large dynamic range) while showing the color as close to the real colors of the scene, as they would have been perceived by a human observer.

Manuscript received 23 January 2024; accepted 25 February 2024. Date of publication 4 March 2024; date of current version 15 March 2024. This work was supported by the Romanian Excellence Center on Artificial Intelligence on Earth Observation Data for Agriculture (AI4AGRI) Project titled "Romanian Excellence Center on Artificial Intelligence on Earth Observation Data for Agriculture" funded by European Union's Horizon Europe Research and Innovation Program under Grant 101079136. *(Corresponding author: Ioana Cristina Plajer.)*

Ioana Cristina Plajer, Alexandra Băicoianu, and Luciana Majercsik are with the Department of Mathematics and Computer Science, Faculty of Mathematics and Computer Science, Transilvania University of Braşov, 500024 Braşov, Romania (e-mail: ioana.plajer@unitbv.ro; a.baicoianu@unitbv.ro; luciana.carabaneanu@unitbv.ro).

Mihai Ivanovici is with the Department of Electronics and Computers, Faculty of Electrical Engineering and Computer Science, Transilvania University of Braşov, 500024 Braşov, Romania (e-mail: mihai.ivanovici@unitbv.ro).

Digital Object Identifier 10.1109/TGRS.2024.3372639

An accurate visualization of satellite images can facilitate a direct human interpretation for spotting pixels or areas affected by various degradation (e.g., in agriculture, spotting areas affected by diseases of the agricultural crops). Moreover, satellite images rendered in a more realistic manner can be used as a first step to verify the accuracy of different approaches, such as classification, which can be initially appreciated visually.

A major challenge in addressing different tasks on MS and HS images is the variability of the image data (spectral range and spatial resolution) when obtained by different sensors. Thus the primary goal of our research, presented in this paper, is a method of processing such aggregated datasets, obtained from various sources, using a fully connected neural network (FCNN). Visualization is merely a secondary goal, as a test case specifically for this fusion.

Although research in the area of visualization is still limited, as already described in [24] and [23], a neural network trained on a set of MS images can be used successfully to visualize spectral images acquired by different sensors. Challenges in this field include limited labeled training data and spectral variability between different sensors, as they capture information at different spectral wavelengths. The spectral variability can make it difficult for a neural network to extract significant and meaningful features in a coherent manner, especially if the training data does not cover the full range of spectral variation.

This article explores different approaches to normalization and standardization, respectively, of the data from different datasets and describes experiments with different preprocessing approaches. Our FCNN methodology efficiently handles both MS and HS images. It also addresses the challenges of band selection and spectral variability, making it a versatile and robust approach for different data settings. Furthermore, the architecture of the network and training hyperparameters are presented in detail.

The main advantages of our method lie in the generalization of the approach and adaptability to data of different spectral and spatial resolutions. Furthermore, by combining publicly-available datasets with appropriately labeled data for network training, we ensure a principled and accessible foundation for model learning.

Our proposed method also has the advantage of not requiring further image postprocessing of the obtained RGB image, which leads to a more polished and refined output straight out of the model.

In the case of FCNN processing, information aggregation is quite beneficial, especially when public data are limited. Such aggregation increases the methodology's efficacy despite related limitations. The visualization results demonstrate potential even when dealing with datasets containing fewer bands, indicating the versatility and robustness of our methodology across various data configurations. Such a methodology eliminates any doubt regarding the selection criteria of the effective bands used.

This article is organized as follows. Section II describes different MS and HS datasets used in our experiments, pointing out the challenges of working with multisource data. Section III presents the data preprocessing steps and the proposed FCNN model architecture, together with the model training process. The visual results of our experiments for different strategies used are presented and discussed in Section IV, and their comparative quality assessment is outlined in Section V, showing the potential of the new approach proposed in Section IV. Finally, the conclusions of this article and some limitations of our study, which became apparent after the completion of the research are emphasized in Section VI, together with future directions for our research.

II. Materials

A. Hyperspectral and Multispectral Imaging

MS and HS images are two types of remote sensing data that capture information about the Earth's surface by sensing electromagnetic radiation. As mentioned in the introduction, they are widely used in various fields, including agriculture and environmental monitoring. The key distinction between MS and HS images lies in their spectral resolution. HS images have a far higher spectral resolution, catching data in several narrow and contiguous bands, while MS images only capture data in a small number of discrete bands. This higher spectral resolution allows for a more detailed characterization of surface materials based on their spectral signatures. MS and HS images offer valuable insights into the Earth's surface, and the choice between them depends on the specific application requirements and the level of spectral and spatial detail needed for analysis.

Differences in sensor types used to capture MS and HS images can introduce specific challenges and variations in the data. Different sensors may have varying spectral band configurations, capturing data in different wavelength ranges or bandwidths. Multiple sources and sensor variety can result in inconsistency when comparing or combining data from different sensors. It is essential to carefully account for these differences to ensure accurate and meaningful analysis. Also, sensors used in MS and HS imaging require calibration to ensure the accuracy and reliability of the captured data. However, calibration procedures can vary between sensors and platforms. Inconsistencies in calibration methods can lead to differences in the radiometric or spectral accuracy of the acquired data. Proper calibration and validation techniques must address these issues and establish reliable and consistent datasets. Each sensor has its characteristics and limitations, such as spectral response functions, signal-to-noise ratios, and dynamic ranges. These characteristics can also affect the quality and reliability of the captured data. In spectral imaging, sensors can have different spatial resolutions, affecting the level of detail captured in the images. Combining or comparing red multisource sensor data with different spatial resolutions can introduce challenges, as the spatial variability may need to align better.

In addition, researchers and analysts should be aware of the sensor-specific characteristics and limitations and consider them when interpreting and comparing data from different sources. Standardization efforts are also conducted to facilitate cross-sensor compatibility and consistency. While multisource sensor differences can introduce challenges and variations in the data, proper preprocessing and understanding of the

sensor characteristics can be addressed to ensure accurate and meaningful MS and HS imagery analyses.

B. Description of the Datasets

The availability of MS and HS imagery with high-resolution spectral information has redefined our understanding of environmental and ecosystem phenomena. These datasets are increasingly accessible for both scientific and practical purposes, and their analysis represents an initial step toward gaining a deeper comprehension of the aforementioned phenomena. Several datasets are recognized and used in this article.

This study selected the CAVE and UGR datasets for the model's training due to their well-established reputation in the literature. They offer a diverse range of colors, and the images were captured under various environmental conditions. Moreover, a corresponding RGB image is available for each image in these datasets, facilitating a comprehensive analysis.

For the testing phase, three HS images were used, two of them provided by the relatively recent PRISMA satellite and one acquired by the Reflective Optics System Imaging Spectrometer (ROSIS-3) over Pavia University, Pavia, Italy.

1) CAVE Dataset: The CAVE dataset [25] comprises 32 MS images captured indoors under controlled illumination conditions. Each image has a resolution of 512 × 512 pixels in the spatial domain. The spectral range is between 400 and 700 nm, with a sampling interval of 10 nm, resulting in a total of 31 spectral bands. A corresponding RGB image with the same spatial resolution is provided for each MS image. This dataset does not contain any natural scenes.

2) UGR Dataset: The UGR dataset [26] contains 14 outdoor images of urban scenes. Most of these images possess a spatial resolution of 1000 × 900 pixels. The spectral range spans from 400 to 1000 nm, with a sampling interval of 10 nm, resulting in 61 spectral bands, out of which 31 fall within the visible spectrum. In addition, each MS image is accompanied by a corresponding RGB image, sharing the same spatial resolution.

As can be observed from the description, the CAVE and the UGR MS images have in common the first 31 wavelengths from the visual domain. This makes them suitable for using the same FCNN with 31 input neurons. From UGR images, we thus considered only the spectral bands in the visual range, as these are relevant for visualization.

3) PRISMA Images: The PRISMA images used for coloring in this study are obtained from the PRISMA HS satellite operated by the Italian Space Agency (ASI). Two specific images were captured, one on October 18, 2022, in the northern region of Brasov county, Romania, and the other on March 24, 2023. The satellite's HS sensors are capable of capturing images within a wide wavelength range of 239 spectral bands, spanning from 400 to 2500 nm. Among these bands, 66 falls within the visible-near infrared range (400–1010 nm), while 173 bands reside in the short-wave infrared range (920–2500 nm). The spectral sampling interval for the satellite's images is less than 12 nm. Regarding spatial characteristics, the images possess a resolution of 1000 × 1000 pixels, with a ground sample distance of 30 m [27].

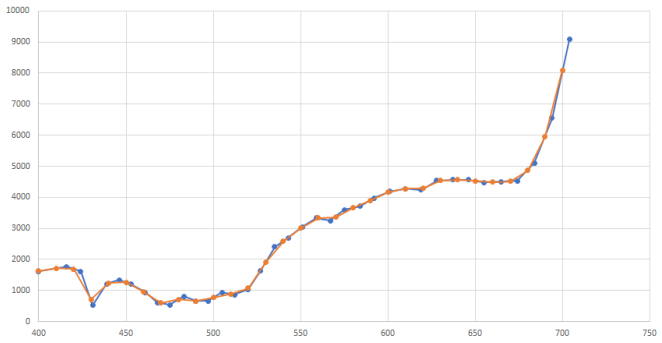

Fig. 1. Original values of one HS pixel from the first PRISMA image (blue) and interpolated values (orange).

The spectral bands used for the experiments are in the visible domain, covering the range from 406 to 713 nm, with an approximate sampling interval of 8 nm.

4) Pavia University Dataset: The Pavia University dataset was made available by the Telecommunications and Remote Sensing Laboratory, Pavia University, in 2001. This dataset was acquired using the ROSIS sensor during a flight campaign conducted over Pavia. The dataset comprises 610 × 610 pixels and covers a wavelength range from 430 to 860 nm with 115 spectral bands. It has a spatial resolution of 1.3 m and a spectral resolution of approximately 4 nm. However, some samples within the image do not contain any useful information and must be eliminated before analysis. Once the broken bands are removed, the 103 remaining bands can be used further in the investigation [28].

In order to test the PRISMA and the Pavia University images on the FCNN trained on the CAVE or the UGR dataset, it is necessary to adapt these images to the network's input layer. As this input is calibrated to receive pixels with the spectral signature of CAVE images, each spectral pixel from the test image has to be mapped on the wavelengths of a CAVE image pixel. This has been done in this study by linear interpolation. For each wavelength of a test image (PRISMA or Pavia University), the value of a CAVE image channel is interpolated from the values of the two neighboring channels of the test image.

Fig. 1 represents the original HS pixel from a PRISMA image together with the interpolated values of this pixel.

It can be seen from Fig. 1 that by linearly interpolating the PRISMA HS image to fit the bands of the CAVE dataset, the changes in the data profile are negligible, thus justifying this approach.

C. Prerequisites

In the context of machine learning experiments, standardization, and normalization are preprocessing techniques used to transform input data into a specific range or distribution. These techniques are commonly applied to improve the performance and convergence of machine learning models [29].

Standardization (z-score normalization or feature scaling) transforms the dataset's features into zero mean and unit variance. It entails dividing each data point by the standard deviation after taking the mean value of the feature out of each data point.

The formula for standardization is shown in the following equation:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

where z is the standardized value, x is the original value, μ is the mean of the feature, and σ is the standard deviation of the feature.

Standardization ensures that each feature has a similar scale and range, which is beneficial for algorithms that assume normally distributed data or when features have different scales. It centers the data around 0, with a standard deviation of 1.

Normalization (min–max scaling) transforms the dataset's features to a standard range—typically between 0 and 1. Still, the range depends on the specific normalization technique used and the requirements of the dataset. It involves subtracting the minimum value of the feature from each data point and then dividing it by the range.

The formula for normalization is shown in the following equation:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2}$$

where x' is the normalized value, x is the original value, $\min(x)$ is the minimum value of the feature, and $\max(x)$ is the maximum value of the feature.

Normalization preserves the shape of the distribution and is suitable when the absolute values of the features are not important but rather their relative values or ratios.

Both standardization and normalization help improve the performance of machine learning models by ensuring that features are on a similar scale and avoiding the dominance of certain features due to their larger magnitudes. It is important to note that the impact of standardization and normalization on performance can vary depending on the dataset and the specific machine learning algorithm used. Therefore, it is advisable to experiment with both techniques and evaluate their impact on the model's performance before finalizing the preprocessing approach.

III. Methods and Procedures

The objective of this Section is to discuss some elements crucial to the success of the tests conducted for this study. For the proposed visualization to perform, it is of great importance to comprehend the preprocessing steps and the model employed.

A. Data Preprocessing Step

The preprocessing stage is essential to data preparation in machine learning and data analysis. Its main objective is to convert raw data into a format suitable for additional analysis or modeling.

Standardization and normalization are two of the many methods included in preprocessing. They seek to improve the performance and efficiency of various algorithms by bringing the data within a predictable and manageable range. Depending on the particular dataset, the type of issue, and the algorithms being applied, these preprocessing methods can change. It is advisable to test several transformation strategies and assess their effects on the model's performance to choose the best approach for a particular task.

In this study, various data preprocessing and transformation variations were tried precisely to study the feasible strategies and related results comprehensively. Thus, two strategies were highlighted in this article.

1) A strategy involving preprocessing on each file, concatenation of data from all the input files, shuffle, and separation by train subset and test subset ((3/4) and (1/4), respectively, of all existing pixels). We will further call this strategy *individual preprocessing*.
2) A strategy involving concatenation without any preprocessing beforehand, shuffle, splitting into train and test (with the exact percentages as the previous strategy), and application on the training subset of one of the proposed preprocessing methodologies (standardization or normalization). After training, our learning algorithm has learned to deal with the data in scaled form, so we have to normalize/standardize our test data with the normalizing/standardizing parameters used for training data. We will further call this strategy *global preprocessing*.

From a technical perspective, for both strategies, *StandardScaler()* [30] and *MinMaxScaler()* [31] have been used for standardization/normalization, respectively.

B. Proposed AI Model Architecture

The problem of consistent spectral image visualization is of significant importance, as it enables the users to visually interpret and understand the acquired data. For sensors like those of Landsat or Sentinel, with a small number of spectral bands overall and three bands in the visual range, the visualization problem is straightforward, as the three bans for red, green, and blue generate the corresponding RGB images. With the upcome of multisource modern MS and HS sensors with a wide range of spectral bands in the visual range, the problem of correct and accurate visualization gets more complex. Generally, classical methods like the ones mentioned in the introduction, often result in low-quality images and usually need adjustments.

Starting from the ideas in [23], an FCNNwas constructed and trained on a set of MS images, as mentioned in [24], in order to visualize spectral images acquired by different sensors. Below, we provide a detailed description of the network architecture and the hyperparameters utilized during training.

1) Model Description: Consistently mapping n-dimensional spectral pixel on a tridimensional RGB pixel can be considered a regression problem, and thus, an FCNN is an appropriate model for learning this mapping. The model, as illustrated in Fig. 2, consists of five layers: an input, an output layer, and three hidden layers.

The input layer consists of 31 neurons, as the number of channels of the CAVE images is 31, all in the visual range, and the number of spectral bands in the visual range of the UGR images is also 31. The number of output neurons is 3, one for each RGB color.

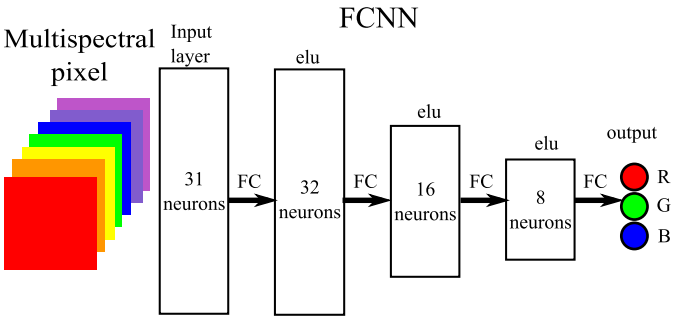

Fig. 2. Model pipeline.

In the hidden layer, the number of neurons was chosen to decrease from the size of the input toward the size of the output by a factor of 2. From this consideration and taking into account the computational advantage of data structured as the power of 2, in the first hidden layer, we used 32 neurons, in the second 16, and in the third 8. The activation function on the hidden layers is the exponential linear unit (ELU) [32], as we did not want to discard negative values.

Assessing machine learning models' performance is crucial for evaluating their effectiveness and suitability for specific assignments. Two commonly used loss functions are mean squared error (MSE) and mean absolute error (MAE). As MSE is very sensitive to outliers, while MAE reduces their impact almost completely, we chose the HuberLoss function, which combines both errors in an balanced way [33]. The Huber loss is defined as follows:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2, & |y - f(x)| \le \delta \\ \delta\left(|y - f(x)| - \frac{1}{2}\delta\right), & |y - f(x)| > \delta \end{cases} \tag{3}$$

where y is the true value, $f(x)$ is the predicted value, and δ is a parameter that determines the threshold for which, in the loss function, the MSE is replaced by MAE.

For the data points where the distance between the label and the predicted value is small, less than the threshold δ, the function behaves like MSE. This makes it more robust to noise. However, for data points with larger differences, it switches to a linear loss. This makes it less sensitive to outliers and helps in providing a more stable estimation. If the outliers comprise 20%–30% of the data, the MAE will ignore them entirely. However, Huber loss will create a balance if the outliers are significant, and therefore, it is a useful choice when dealing with datasets that may contain outliers or noise.

The selection of δ plays a crucial role. In our case, the value $\delta = 10.0$ parameter for the HuberLoss was selected empirically. We considered that a difference of 10 between two values (of a color channel) is acceptable for using the MSE, similar to just noticeable difference (JND) corresponding to $\Delta E = 3$ in CIELab.

2) Model Training: After substantial testing, 150 epochs for training were determined to be adequate for accurate color mapping. The number of pixels in a batch was 2048 to shorten training time. In the backpropagation step, a common value for the learning rate $\alpha = 0.005$ and the Adam optimizer provided by the Pytorch library was used.

The model was trained on Intel[1] Core[2] i7-7700 CPU at 3.60-GHz server with eigh CPUs, and the duration of the training on 150 epochs on the UGR dataset was approximately 2 h.

The set used for training was preprocessed, as discussed in Section IV-A2, by concatenating all the pixels from all the images, shuffling, partitioning them into train and test sets, and standardizing over the train set.

The training algorithm can be described as follows.

Model Training Steps

1) Load pixels from all the images of the dataset (CAVE or UGR), shuffle the pixels, and partition into train and test (75% versus 25%).
2) Preprocess data by standardization using scikit-learn StandardScaler [30].
3) Split set in random batches of 2048 pixels.
4) Train the model using HuberLoss.

In order to enhance clarity and facilitate a deeper understanding of the proposed methodology, pseudocode has been provided to describe key steps in detail in Algorithm 1.

Algorithm 1 Pseudocode for the Training of the FCNN

Require: $path\_dataset$, $model$, $lr = 0.005$, $epochs = 150$
$(data, labels) \leftarrow load\_pixels(path\_dataset)$
$shuffle(data, labels)$
$nr\_train\_data \leftarrow (\frac{3}{4}) * len(data)$
$nr\_test\_data \leftarrow len(data) - nr\_train\_data$
$train\_set \leftarrow (data.head(nr\_train\_data), labels.head(nr\_train\_data))$
$test\_set \leftarrow (data.tail(nr\_test\_data), labels.tail(nr\_test\_data))$
Initialize the StandardScaler and Standardize training data
Apply Scaler on test data
$batch\_size \leftarrow 2048$
Initialize weights of the *model*
$optimizer \leftarrow AdamOptimizer(lr, weight\_decay = 0.0008)$
$best\_loss \leftarrow maxValue$
$loss\_function \leftarrow HuberLoss(10.0)$
$train\_loss\_vec \leftarrow []$
$test\_loss\_vec \leftarrow []$
for $epoch \in (1, epochs)$ **do**
  $train\_loss \leftarrow 0$
  for $batch, (x, y) \in train\_set$ **do**
    $y\_pred = model(x)$
    $loss \leftarrow loss\_function(y, y\_pred)$
    Backpropagation Step - backprop(loss)
    $train\_loss \leftarrow train\_loss + loss.item()$
  end for
  $train\_loss \leftarrow train\_loss/(batch\_size)$
  $train\_loss\_vec.append(train\_loss)$
  if $train\_loss < best\_loss$ **then**
    Save current weights
    $best\_loss \leftarrow train\_loss$
  else
  end if ▷ Now evaluate on test set
  $test\_loss \leftarrow 0$
  for $batch, (x, y) \in test\_set$ **do**
    $y\_pred \leftarrow model(x)$
    $loss \leftarrow loss\_function(y, y\_pred)$
    $test\_loss \leftarrow test\_loss + loss.item()$
  end for
  $test\_loss \leftarrow test\_loss/(batch\_size)$
  $test\_loss\_vec.append(test\_loss)$
end for
Save current weights
$plot(train\_loss\_vec, test\_loss\_vec)$

The graphical representation of the loss decay on the train and the test sets for the CAVE and UGR datasets are represented in Fig. 3. It can be seen from this figure, that

[1]Registered trademark.
[2]Trademarked.

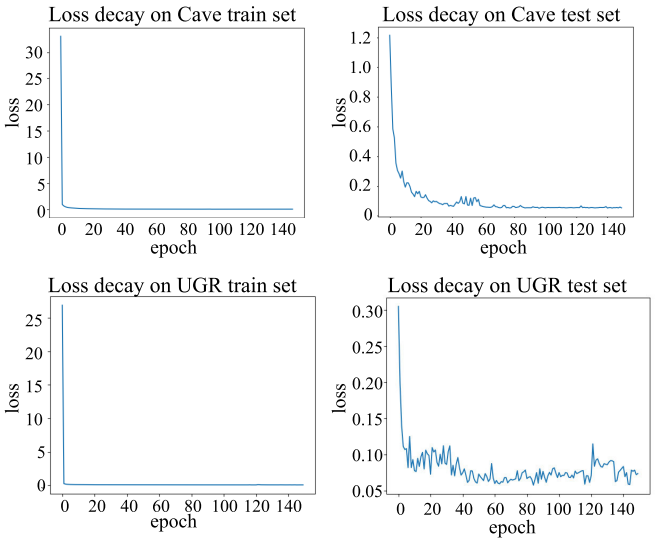

Fig. 3. Loss decay on the train and test sets when training the model with the CAVE dataset and UGR dataset, respectively.

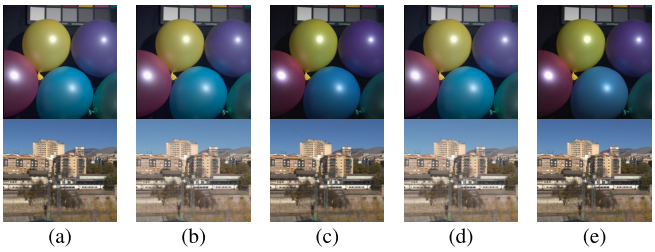

Fig. 4. Visualization results of the experiments described in Section IV-A, in which global preprocessing was done on the datasets. (a) One sample of CAVE and UGR datasets, respectively. The other subfigures show the results of the FCNN trained on (b) CAVE and (c) UGR, respectively, with global normalization and on (d) CAVE and (e) UGR, respectively, with global standardization of the training set.

on the train set, the loss decays very steeply during the first epochs and then gets stabilized, while to achieve similar results on the test set, the network needs more training.

IV. Results for Different Strategies

A. Results With Global Preprocessing

In these experiments, the general strategy is that after splitting into the train and test subsets, normalization or standardization was performed on the entire training set, and then, the results were applied to the test set. For inference, the chosen method is applied to the image being colored, using the parameters from the train.

1) Global Preprocessing With Normalization: Fig. 4(b) presented the results of the FCNN trained on the CAVE dataset, and Fig. 4(c) presented the results when training the FCNN on the UGR dataset, on one sample image, see Fig. 4(a), from each of these datasets. The data was preprocessed by min–max normalization on the respective training set using the MinMax scaler.

The results of the FCNN trained on the CAVE and UGR Datasets on the Pavia University image are presented in Fig. 6(a) and (b). The results of the model on one of the images acquired by the PRISMA Satellite in both treating

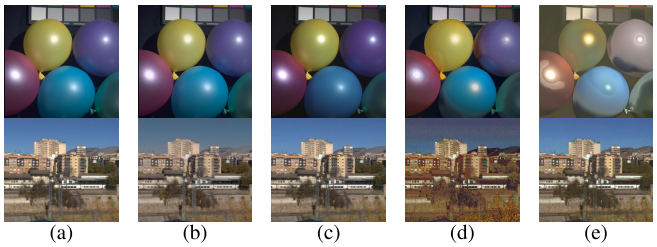

Fig. 5. Visualization results of experiments described in Section IV-B, in which preprocessing was done on each image of the datasets separately. (a) One sample of CAVE and UGR datasets, respectively. The other subfigures show the results of the FCNN trained on (b) CAVE and (c) UGR, respectively, with normalization and on (d) CAVE and (e) UGR, respectively, with standardization on each image of the training set.

scenarios are presented in Fig. 7(a) and (b). The Pavia University and the PRISMA spectral images were interpolated, to match the input spectral bands and were normalized using the parameters obtained on the respective training sets.

2) Global Preprocessing With Standardization: The results of the FCNN trained separately on the CAVE and the UGR datasets, respectively, on one sample image of each of these datasets [Fig. 4(a)] are presented in Fig. 4(d) for the CAVE and in Fig. 4(e) for the UGR trained network. The standardization in each case was done on the corresponding training dataset using the standard scaler.

The results of this method on the Pavia University image can be seen from Fig. 6(c) and (d), and the results on PRISMA HS image are presented in Fig. 7(c) and (d).

B. Results With Individual Preprocessing

In these experiments, before concatenating all the images and separating them into train and test subsets, each image is taken and brought into the same range of values by normalizing/standardizing each one according to its own values.

1) Individual Preprocessing by Normalization: Each image is considered separately, and then normalized according to the min/max in the image, thus transforming all values into the range [0, 1]. All the images are then concatenated and the resulting pixel set is split into train and test subsets, respectively. At inference, an image is normalized and colored according to its associated min/max values.

Fig. 5 presented the results of the coloring using the FCNN trained on CAVE and UGR datasets, respectively, normalizing each image before concatenating with the others.

Fig. 6(e) and (f) presented the coloring results on Pavia University and Fig. 7(e) and (f) those on the PRISMA image, with the CAVE, respectively UGR-trained FCNN, normalizing each image with respect to its own min–max values.

2) Individual Preprocessing by Standardization: In this approach, each image is considered separately, and then standardized according to the mean/average deviation in the image, so all values are brought into the standard range $[-1, 1]$. The processed images are then concatenated and the resulting pixel set is split into train and test subsets, respectively. At inference, an image is standardized and colored according to the distribution of values in the image.

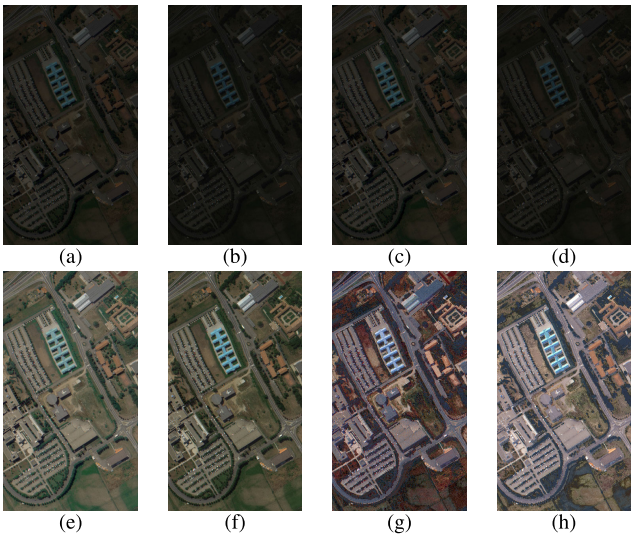

Fig. 6. Visualization of the results on the Pavia University MS image using the FCNN trained in the described scenarios on (a) CAVE and (b) UGR, respectively, with global normalization and using for inference the same scaler as for the training set; on (c) CAVE and (d) UGR, respectively, with global standardization and using for inference the same scaler as for the training set; on (e) CAVE and (f) UGR, respectively, with normalization of each image with respect to its own min–max values; and on (g) CAVE and (h) UGR, respectively, with standardization of each image with respect to its own mean and standard deviation.

Fig. 5 presented the results of the coloring using the FCNN trained on CAVE and UGR datasets, respectively, standardizing each image before concatenating with the others, on samples of these datasets.

Fig. 6(g) and (h) presented the coloring results on Pavia University and PRISMA using the CAVE- and UGR-trained FCNN, respectively, normalizing each image with respect to its own mean/std values. The visualization results of the FCNN on a PRISMA HS image are presented in Fig. 7(g) and (h).

C. Different Approach

The experiments interpreted from the previous figures show that the best coloring results on the CAVE and UGR datasets were obtained when normalizing or standardizing the training set (global preprocessing) and using the respective parameters on the test set. This strategy follows the standard practice in machine learning flows. On the other hand, the results were not acceptable on the Pavia University and the PRISMA images. This is not surprising, as these images are bound to.

Fig. 8 illustrated the distributions on the green channels (650 nm) for the CAVE Fig. 8(a) and the UGR Fig. 8(b) datasets, together with those of Pavia University Fig. 8(c) and the first PRISMA image Fig. 8(d), both calculated after interpolation. We selected the green channel for illustration, but the observation is valid for any other 31 channels of the spectral images considered.

As we aimed to obtain a model that should be able to visualize images from different acquiring sources with different distributions and spectral signatures, we tried following a nonstandard approach. We considered the FCNN model, with global preprocessing by standardization in the training stage. After training, to visualize Pavia University and PRISMA images, these were standardized concerning their own mean and standard deviation to obtain for each channel the mean of 0 and the standard deviation of 1.

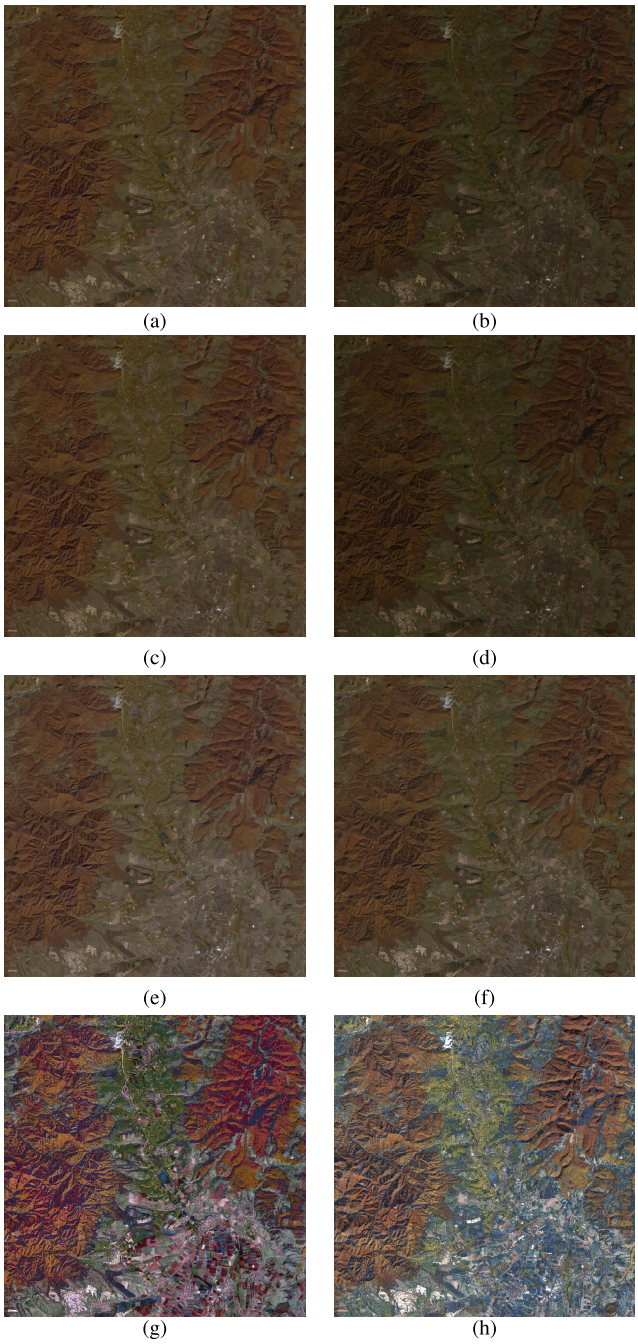

Fig. 7. Visualization of the first PRISMA HS image using the FCNN trained in the described scenarios on (a) CAVE and (b) UGR, respectively, with global normalization and using for inference the same scaler as for the training set; on (c) CAVE and (d) UGR, respectively, with global standardization and using for inference the same scaler as for the training set; on (e) CAVE and (f) UGR, respectively, with normalization of each image with respect to its own min–max values; and on (g) CAVE and (h) UGR, respectively, with standardization of each image with respect to its own mean and standard deviation.

The following algorithm can express the inference step.

Visualization Algorithm

1) Interpolate image to fit CAVE spectral range.
2) Load the pixels of the interpolated image.
3) Standardize the pixels with PyTorch StandardScaler relative to their mean and variance.

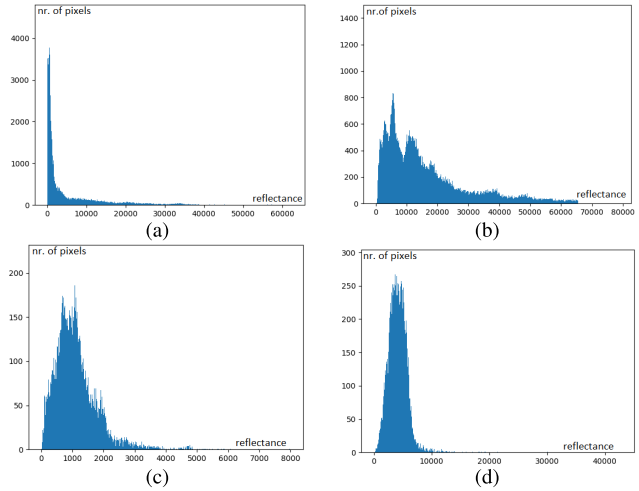

Fig. 8. Distributions on the green channel (considered at 550 nm) for CAVE and UGR datasets and for Pavia University and PRISMA spectral images. (a) Distribution for CAVE. (b) Distribution for UGR. (c) Distribution for Pavia University. (d) Distribution for PRISMA.

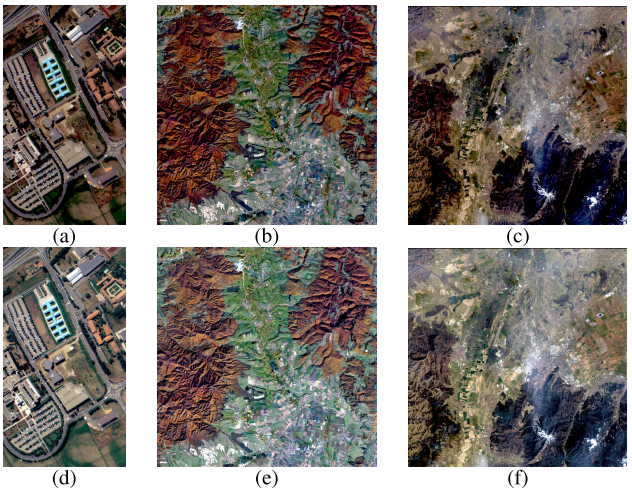

Fig. 9. Visualization of the results on Pavia University (a), first PRISMA image (b) and second PRISMA image (c) when standardizing those images according to their own distribution using the CAVE trained FCNN. Visualization of the results on Pavia University (d), first PRISMA image (e) and second PRISMA image (f) when standardizing those images according to their own distribution using the UGR trained FCNN.

4) Use the model to predict the corresponding (R, G, B) triplet for each interpolated image pixel.
5) Construct the RGB image with respect to the original size of the input image.

To enhance clarity and promote a thorough understanding of the proposed methodology, we have incorporated pseudocode of Algorithm 2 to provide detailed descriptions of inference step.

Algorithm 2 Pseudocode for Visualization using the FCNN

Require: $path\_ms\_image$, $model$, $input\_freq$, $target\_freq$
1: $image \leftarrow load(path\_ms\_image)$
2: $interpolated \leftarrow$ Interpolate$(image, input_freq, target_freq)$
3: Scale $interpolated$ with StandardScaler
4: $predicted \leftarrow model(interpolated)$
5: Make RGB image from $predicted$

The results of this approach are presented as follows: for the Pavia University image in Fig. 9(a) (CAVE-trained FCNN) and 9(d) (UGR-trained FCNN), for the first PRISMA image in Fig. 9(b) (CAVE-trained FCNN) and 9(e) and for the second PRISMA image in Fig. 9(c) and (f), respectively.

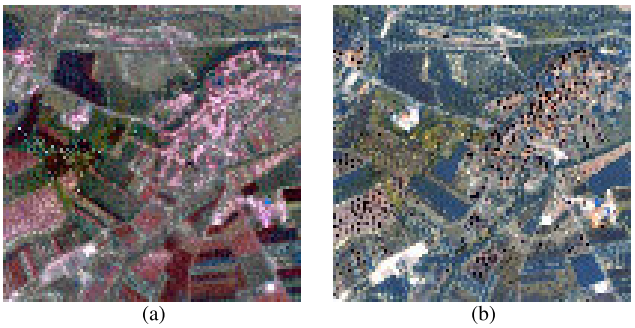

Fig. 10. Illustrations of the artifacts in the PRISMA image in a region of interest. (a) Region from the image in Fig. 7(g). (b) Region from image in Fig. 7(h).

V. Discussion and Comparisons

Various experiments have been carried out in this study, all aiming to better visualize the considered spectral images. And since we want to determine which of the discussed approaches is better, a variety of data from different types of sensors were used.

All the results to be commented on are presented from Figs. 4–13. We will comment on each experiment, in the same order in which they were performed.

Fig. 4 shows the results of the experiments in the case of global preprocessing by normalization over the whole training set. As can be noticed, visually, the results are quite similar in terms of colors. However, on closer inspection, it can be seen that there are also pairs with better contrast, in the sense that the coloring of an UGR image works better with a network trained with UGR images and similarly for CAVE images.

Fig. 5 shows the results of the experiments when each image is normalized/standardized according to its own values. As can be seen, normalization works better than standardization, which makes sense because during the preprocessing by normalization each feature undergoes a transformation to fit within a new range, while preserving its original relationships with the other features in the data, meaning that all the relational properties within the data remain intact [34].

Following these tests, this type of preprocessing with normalization seems to be a consistent option. The alternative with standardization is not plausible because this transformation step tends to change the relationships between colors, which leads to artifacts, especially for images that are not part of the training dataset, see Fig. 10(a) and (b).

Figs. 6 and 7 represent the results of tests performed on images obtained from different types of sensors and having different characteristics than the images in the training sets. To use the network on images such as Pavia University image or the images acquired by PRISMA, a mandatory step was the interpolation one, which is necessary to map the spectral bands of those images on the input with 31 channels of the model.

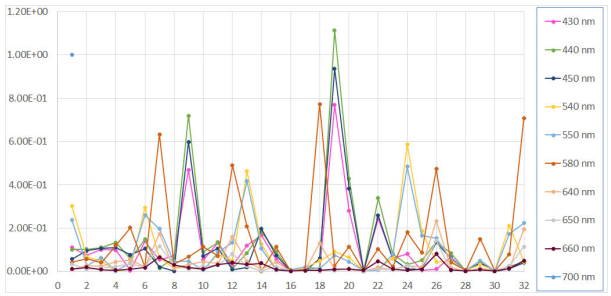

Fig. 11. Selection of neurons with highest absolute values on the first layer.

Regardless of the training dataset, CAVE or UGR, if we normalize or standardize over the whole training set and at the inference step we use the scaler/properties from the training stage, the results for the images taken from other sensors are very bad in terms of brightness and contrast, on Pavia even worse than on PRISMA [see Fig. 6(d)—standardization on UGR versus Fig. 7(d)].

Normalization on each image yields good results on Pavia [Fig. 6(e) and (f)] and acceptable on PRISMA [Fig. 7(e) and (f)]. In contrast, standardization on each image generates significant artifacts, see Figs. 6(g) and 7(g) for the standardization on each image in CAVE, and Figs. 6(h) and 7(h) for the standardization on each image in UGR. These artifacts can be seen more clearly on a selected region in PRISMA image, as illustrated in Fig. 10.

In Fig. 9, we presented different experiments for visualization of spectral images by means of a FCNN. As could be seen from all previous figures compared with Fig. 9, the best results on images that were acquired by other sensors than those used for the images in the training set and with another spectral signature, were obtained by using global preprocessing in the training step, but standardizing these images according to their own mean and standard deviation at inference.

We also compared our results with some conventional methods of spectral image visualization namely band selection and XYZ space [35] as well as two other methods: decolorization-based HSI visualization [36] and multichannel pulse-coupled neural network (MPCNN)-based HSI visualization [37]. Figs. 12 and 13 displayed the results of the visualization of the first PRISMA image and the second PRISMA image, respectively, using these methods.

Moreover, we studied if our method favors certain wavelengths with respect to others, as is the case for band selection. By plotting the weights between the input and the first hidden layer, we found out, that all the bands contribute in a balanced manner to the final results. Fig. 11 displayed a selection of these weights. Each color represents the weights of one input neuron to the neurons on the next layer, while each input neuron corresponds to one wavelength.

In conclusion, the results of our tests are better than the results of conventional methods, at least on PRISMA, our use case, regardless of how the preprocessing stage is performed. Given the fact that the standardization for other types of images is done relative to their mean and variance, we believe that a network that has been trained on images with a good distribution of light and contrast will tend to produce images

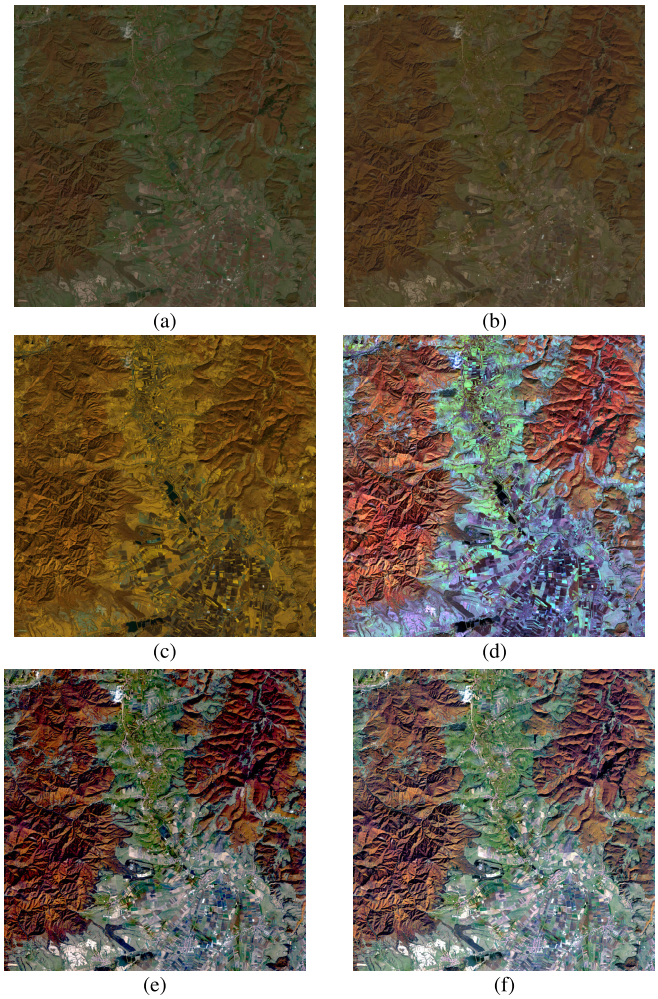

Fig. 12. Comparative results of classic and advanced visualization techniques with the new approach for the first PRISMA image. (a) Band selection. (b) Coloring with XYZ space. (c) Decolorization-based visualization. (d) MPCNN visualization. (e) Results with CAVE-trained FCNN. (f) UGR-trained FCNN.

with good contrast and light, even if the original images are affected by the atmospheric conditions. These results justify using this type of approach in the case of satellite images.

A. Comparative Quality Assessment

In order to perform a numerical quality comparison between the results presented in Figs. 12 and 13, some no-reference image quality assessments [38], [39] like entropy, which estimates information quantity, fractal dimension, which estimates the complexity and standard deviation, which estimates nonuniformity, were used. These values were calculated using adequate scripts offered by Ivanovici and Richard [40] for entropy and Ivanovici [41] and Caliman et al. [42] for fractal dimension, and the MATLAB *std2* function for the standard deviation of an image. The PRISMA images were correspondingly scaled as the scripts were calibrated for 256×256. For calculating the fractal dimension, the parameters used were $\text{LMAX} \in \{41, 71, 101\}$, representing the maximum size of the hypercubes and a threshold of 0.00001 for the standard deviation [43].

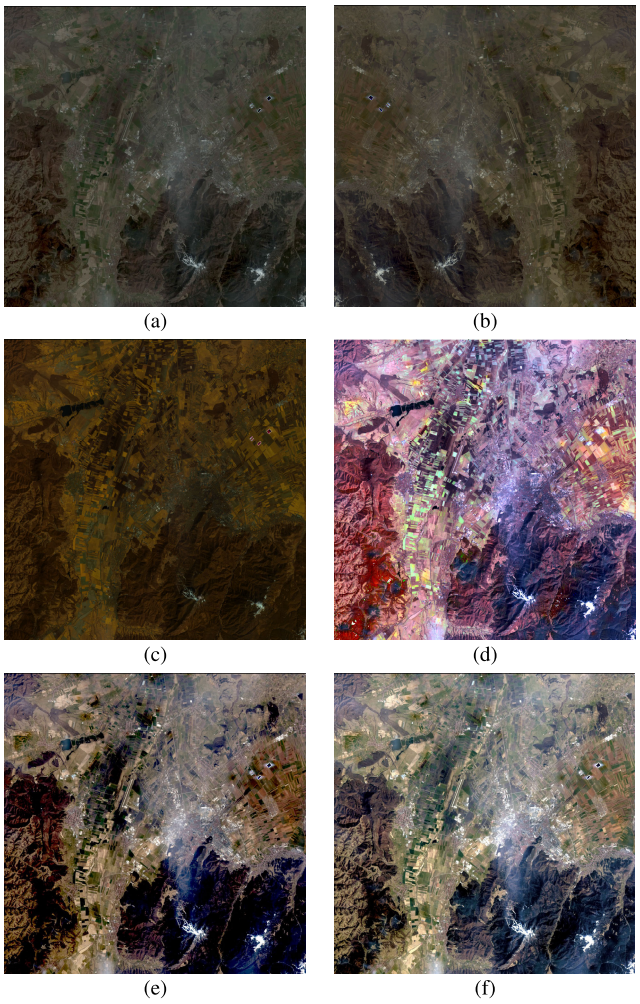

Fig. 13. Comparative results of classic and advanced visualization techniques with the new approach for the second PRISMA image. (a) Band selection. (b) Coloring with XYZ space. (c) Decolorization-based visualization. (d) MPCNN visualization. (e) Results with CAVE-trained FCNN. (f) UGR-trained FCNN.

It is important to note that the actual entropy values of color images will depend on the specific content and distribution of pixel intensities in the image. Images with complex color patterns, a wide range of intensities, and diverse color distributions are likely to have higher entropy values. Hence, a higher value of the entropy indicates a higher informational content, and a higher value of the fractal dimension indicates higher complexity of the scene and is directly correlated with the contrast of the image. A larger variance is also characteristic of images with a larger variety, thus greater informational content.

When satellite land cover images include agricultural areas, as well as urban and nonagricultural surfaces, as seen in the PRISMA images presented in this study, we expect a consistent visualization to exhibit large values for the entropy, the fractal dimension, and the variance.

The processed values for these measures in the case of the visualizations presented in Figs. 12 and 13 are displayed in Tables I and II. It can be observed that the largest values for entropy and variance are obtained in the case of the MPCNN method, closely followed by the results

TABLE I
COMPARATIVE QUALITY ASSESSMENT FOR THE FIRST PRISMA IMAGE

Sample	***Entropy***	***Standard Deviation***	***Fractal Dimension with*** L_{max}		
			41	71	101
FCNN / CAVE	15.6314	47.0056	3.7421	3.8271	3.8291
FCNN / UGR	15.8624	44.7105	3.7199	3.8064	3.7940
Band selection	12.6552	16.1236	2.7463	2.7484	2.8136
XYZ method	13.5262	20.4681	2.9795	2.9251	2.8930
Decolorization	14.0430	39.5901	2.9668	3.0568	3.1282
MPCNN	15.9284	48.0310	3.7245	3.9921	3.9566

TABLE II
COMPARATIVE QUALITY ASSESSMENT FOR THE SECOND PRISMA IMAGE

Sample	***Entropy***	***Standard Deviation***	***Fractal Dimension with*** L_{max}		
			41	71	101
FCNN / CAVE	14.9011	50.5844	3.1309	3.1925	3.2909
FCNN / UGR	15.5923	48.5822	3.0698	3.1678	3.2157
Band selection	14.2838	27.2990	2.4924	2.7755	3.1075
XYZ method	13.7575	20.7137	2.6343	2.7082	2.8104
Decolorization	12.9841	23.2493	2.6714	2.7646	2.8926
MPCNN	15.8832	53.4223	3.2908	3.4603	3.6736

provided by our FCNN. As the visualization results presented in Figs. 12(d) and 13(d), the MPCNN method provides images with large variability, but with completely unnatural colors, being thus unusable in the case of the PRISMA images. Thus, the visualizations obtained by our method, exhibit the best visual results and almost the best quantitative ones. Consequently, it is evident that the FCNN visualizations yield the best results for all the measures, with slightly greater contrast for the CAVE-trained FCNN. These quantitative results further justify the approach proposed in this article.

VI. CONCLUSION

Processing multisource spectral images is still challenging as more satellites with different sensor characteristics are launched, and their products are freely available for scientists and other users. This article aims to resolve part of the problems posed by a consistent and qualitative visualization of such images, using an FCNN trained on two of the most known public spectral datasets appropriate for this purpose.

Furthermore, we studied and performed several preprocessing procedures, which are very important in this FCNN approach to visualize multisource images exhibiting different spectral signatures.

The results we obtained were evaluated both visually and by conventional measures of information content. This evaluation has shown that the images generated by our method can provide a deeper understanding of various aspects, such as identifying agricultural patches and urban centers.

The visual and quantitative results indicate that the proposed methodology is a promising direction for a consistent and qualitative multisource spectral image visualization. There are still some limitations of this approach. For example, in cases where the source image exhibits minor variance, the application of standardization may compromise the realism of coloring.

The reliance on labeled data for training limits our dataset choices to those that are annotated and publicly accessible. This limitation could affect the diversity and representativeness of the training data; hence, further research is needed to address these issues. In addition, when combining datasets, we must either interpolate or truncate information to balance the dataset differences, each option having its trade-offs.

Our future research efforts will be based on addressing some of these limitations. This preliminary work can prepare the ground for our next objective that is interpreting the data through the perspective of vegetation indices and other elements relevant to the agricultural sector, thus increasing the usefulness of our approach in this sector.

ACKNOWLEDGMENT

The hyperspectral images from the PRISMA satellite presented in this article were kindly provided by the Italian Space Agency (ASI), Rome, Italy.

REFERENCES

[1] M. Weiss, F. Jacob, and G. Duveiller, "Remote sensing for agricultural applications: A meta-review," *Remote Sens. Environ.*, vol. 236, Jan. 2020, Art. no. 111402.

[2] T. Adão et al., "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.

[3] P. Rodríguez-Veiga et al., "Forest biomass retrieval approaches from Earth observation in different biomes," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 77, pp. 53–68, May 2019.

[4] E. Vangi et al., "The new hyperspectral satellite PRISMA: Imagery for forest types discrimination," *Sensors*, vol. 21, no. 4, p. 1182, Feb. 2021.

[5] L. Du et al., "A comprehensive drought monitoring method integrating MODIS and TRMM data," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 23, pp. 245–253, Aug. 2013.

[6] Y.-T. Chan, S.-J. Wang, and C.-H. Tsai, "Real-time foreground detection approach based on adaptive ensemble learning with arbitrary algorithms for changing environments," *Inf. Fusion*, vol. 39, pp. 154–167, Jan. 2018.

[7] X. Kang, X. Zhang, S. Li, K. Li, J. Li, and J. A. Benediktsson, "Hyperspectral anomaly detection with attribute and edge-preserving filters," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5600–5611, Oct. 2017.

[8] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[9] M. Mehmood, A. Shahzad, B. Zafar, A. Shabbir, and N. Ali, "Remote sensing image classification: A comprehensive review and applications," *Math. Problems Eng.*, vol. 2022, pp. 1–24, Aug. 2022.

[10] W. Rao, L. Gao, Y. Qu, X. Sun, B. Zhang, and J. Chanussot, "Siamese transformer network for hyperspectral image target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5526419.

[11] L. Zhuang, L. Gao, B. Zhang, X. Fu, and J. M. Bioucas-Dias, "Hyperspectral image denoising and anomaly detection based on low-rank and sparse representations," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2020, Art. no. 5500117.

[12] L. Gao, D. Wang, L. Zhuang, X. Sun, M. Huang, and A. Plaza, "BS3LNet: A new blind-spot self-supervised learning network for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5504218.

[13] L. Gao, X. Sun, X. Sun, L. Zhuang, Q. Du, and B. Zhang, "Hyperspectral anomaly detection based on chessboard topology," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5505016.

[14] X. Sun, L. Zhuang, L. Gao, H. Gao, X. Sun, and B. Zhang, "Information retrieval with chessboard-shaped topology for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5514515.

[15] S. Le Moan, A. Mansouri, Y. Voisin, and J. Y. Hardeberg, "A constrained band selection method based on information measures for spectral image color visualization," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 5104–5115, Dec. 2011.

[16] H. Su, Q. Du, and P. Du, "Hyperspectral image visualization using band selection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2647–2658, Jun. 2014.

[17] Y. Zhu, P. K. Varshney, and H. Chen, "Evaluation of ICA based fusion of hyperspectral images for color display," in *Proc. 10th Int. Conf. Inf. Fusion*, Jul. 2007, pp. 1–7.

[18] V. Tsagaris, V. Anastassopoulos, and G. A. Lampropoulos, "Fusion of hyperspectral data using segmented PCT for color representation and classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 10, pp. 2365–2375, Oct. 2005.

[19] H. A. Khan, M. M. Khan, K. Khurshid, and J. Chanussot, "Saliency based visualization of hyper-spectral images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 1096–1099.

[20] D. Liao, S. Chen, and Y. Qian, "Visualization of hyperspectral images using moving least squares," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2851–2856.

[21] P. Duan, X. Kang, and S. Li, "Convolutional neural network for natural color visualization of hyperspectral images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 3372–3375.

[22] R. Tang, H. Liu, J. Wei, and W. Tang, "Supervised learning with convolutional neural networks for hyperspectral visualization," *Remote Sens. Lett.*, vol. 11, no. 4, pp. 363–372, Apr. 2020.

[23] R.-M. Coliban, M. Marincaǧ, C. Hatfaludi, and M. Ivanovici, "Linear and non-linear models for remotely-sensed hyperspectral image visualization," *Remote Sens.*, vol. 12, no. 15, p. 2479, Aug. 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/15/2479

[24] I. C. Plajer, A. Baicoianu, and L. Majercsik, "AI-based visualization of remotely-sensed spectral images," in *Proc. Int. Symp. Signals, Circuits Syst. (ISSCS)*, Jul. 2023, pp. 1–4.

[25] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, "Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2241–2253, Sep. 2010.

[26] J. Eckhard, T. Eckhard, E. M. Valero, J. L. Nieves, and E. G. Contreras, "Outdoor scene reflectance measurements using a Bragg-grating-based hyperspectral imager," *Appl. Opt.*, vol. 54, no. 13, pp. 15–24, 2015.

[27] ASI. (Mar. 2020). *Prisma Products Specification Document Issue 2.3*. [Online]. Available: https://prisma.asi.it/missionselect/docs/PRISMA%20Product%20Specifications_Is2_3.pdf

[28] M. Garana, M. A. Veganzones, and B. Ayerdi. (May 3, 2023). *Hyperspectral Remote Sensing Scenes*. [Online]. Available: https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

[29] S. Ozdemir and D. Susarla, *Feature Engineering Made Easy: Identify Unique Features From Your Dataset in Order To Build Powerful Machine Learning Systems*. Birmingham, U.K.: Packt Publishing, 2018.

[30] (Jun. 2023). *Standard Scaler*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[31] (May 2023). *MinMax Scaler*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[32] S. Singh. (Feb. 2023). *ELU As an Activation Function in Neural Networks*. [Online]. Available: https://deeplearninguniversity.com/elu-as-an-activation-function-in-neural-networks/

[33] A. Jung, *Machine Learning: The Basics*. Berlin, Germany: Springer, 2022. [Online]. Available: https://books.google.com/books/about/Machine_Learning.html?hl=it&id=1IBaEAAAQBAJ

[34] G. Aksu, C. O. Güzeller, and M. T. Eser, "The effect of the normalization method used in different sample sizes on the success of artificial neural network model," *Int. J. Assessment Tools Educ.*, vol. 6, no. 2, pp. 170–192, Jul. 2019.

[35] M. Magnusson, J. Sigurdsson, S. E. Armansson, M. O. Ulfarsson, H. Deborah, and J. R. Sveinsson, "Creating RGB images from hyperspectral images using a color matching function," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Sep. 2020, pp. 2045–2048.

[36] X. Kang, P. Duan, S. Li, and J. A. Benediktsson, "Decolorization-based hyperspectral image visualization," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4346–4360, Aug. 2018.

[37] P. Duan, X. Kang, S. Li, and P. Ghamisi, "Multichannel pulse-coupled neural network-based hyperspectral image visualization," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2444–2456, Apr. 2020.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[39] S. Winkler, "Vision models and quality metrics for image processing applications," Ph.D. dissertation, Ecole Politechnique Federale de Lausanne, Lausanne, Switzerland, 2001.

[40] M. Ivanovici and N. Richard, "Entropy versus fractal complexity for computer-generated color fractal images," in *Proc. 4th CIE Expert Symp. Colour Vis. Appearance*, Prague, Czech Republic, 2016, pp. 6–7.
[41] M. Ivanovici, "Fractal dimension of color fractal images with correlated color components," *IEEE Trans. Image Process.*, vol. 29, pp. 8069–8082, 2020.
[42] A. Caliman, M. Ivanovici, and N. Richard, "Colour fractal dimension for psoriasis image analysis," in *Proc. Signal Process. Appl. Math. Electron. Commun. (SPAMEC), EURASIP, Session VI*, Aug. 2021, pp. 113–116.
[43] M. Ivanovici and N. Richard, "Fractal dimension of color fractal images," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 227–235, Jan. 2011.

Ioana Cristina Plajer received the B.E. and M.S. degrees in computer science from the University of Bucharest, Bucharest, Romania, in 1997 and 1998, respectively, and the Ph.D. degree in computer science from the Transilvania University of Braşov, Braşov, Romania, in 2011.

She is currently a Lecturer with the Faculty of Mathematics and Computer Sciences, Transilvania University. She is also a member of the Department's Machine Learning Research Group, founded in 2018 and part of the Project Artificial Intelligence and Earth Observation for Romania's Agriculture (AI4AGRI). Her research interests include machine learning, image processing, spectral imaging and remote sensing, formal languages, algorithms, and data structures.

Alexandra Băicoianu received the Ph.D. degree from Babes Bolyai University, Cluj-Napoca, Romania, in 2016.

She has been a Lecturer with the Transilvania University of Braşov, Braşov, since 2017, teaching various courses and seminars. She is currently a Research Engineer in informatics. She authored or coauthored more than 30 scientific articles and is the coauthor of six books. Also, she has supervised tens of graduation and dissertations thesis, programming training courses, programming Summer Schools, and code/tech Camps, some of them in collaboration with IT companies. She is also a member of the Department's Machine Learning Research Group, founded in 2018. She was part of various scientific projects, among them it is important to mention Advanced Technologies for Intelligent Urban Electric Vehicles, Powerful Advanced N-level Digital Architecture (PANDA), Intelligent Motion Control under Industry4.E (IMOCO4E), Artificial intelligence and Earth observation for Romania's agriculture (AI4AGRI), Digital Technologies and Artificial Intelligence (AI) solutions projects (DiTArtIS), and New modular Electrical Architecture and Digital Platform to Optimise Large Battery Systems on SHIPs (NEMOSHIP). Her research interests and expertise are in the field of machine learning, formal languages and compilers, algorithms, remote sensing and Earth observation data, autonomous driving, and electric and hybrid vehicles.

Luciana Majercsik received the B.S. degree in mathematics from the University of Bucharest, Bucharest, Romania, in 1998. She is currently pursuing the Ph.D. in computer science with the Transilvania University of Braşov, Braşov, Romania.

Her research interests include machine learning, multi-/hyperspectral image analysis and visualization, and graph-based methods in remote sensing.

Mihai Ivanovici (Member, IEEE) received the Ph.D. degree in electronics and telecommunications from Politehnica University, Bucharest, România, in 2006.

He was an Invited Researcher in 2008, 2010, and 2014; and a Visiting Professor with the University of Poitiers, Poitiers, France, in 2018, University Toulouse 3 Paul Sabatier, Toulouse, France, in 2019, and Technical University of Moldova, Chişinău, Moldova, in 2023. He is currently a Full Professor with the Electronics and Computers Department, Transilvania University of Braşov, Braşov, România. He is also the Head of Multispectral Imaging and Vision Research Laboratory. His research interests include in the field of algorithms and electronic system design for signal and data acquisition, processing, and analysis—including color, multi- and hyperspectral images, remote sensing, and Earth observation data, as well as data from particle detectors in the ATLAS Experiment, CERN, Geneva, Switzerland.

Dr. Ivanovici has been a member of the IEEE Signal Processing Society since 2008 and the IEEE Geoscience and Remote Sensing Society since 2018.