AN UNSUPERVISED LABELING APPROACH FOR HYPERSPECTRAL IMAGE CLASSIFICATION

Jonathan González Santiago, Fabian Schenkel, Wolfgang Gross, Wolfgang Middelmann

Fraunhofer IOSB
Gutleuthausstr. 1, 76275 Ettlingen, Germany

KEY WORDS: Hyperspectral Imaging, Segmentation, Superpixel, Hierarchical Clustering, Fuzzy C-Means, Convolutional Neural Networks

ABSTRACT:

The application of hyperspectral image analysis for land cover classification is mainly executed in presence of manually labeled data. The ground truth represents the distribution of the actual classes and it is mostly derived from field recorded information. Its manual generation is ineffective, tedious and very time-consuming. The continuously increasing amount of proprietary and publicly available datasets makes it imperative to reduce these related costs. In addition, adequately equipped computer systems are more capable of identifying patterns and neighbourhood relationships than a human operator. Based on these facts, an unsupervised labeling approach is presented to automatically generate labeled images used during the training of a *convolutional neural network (CNN)* classifier. The proposed method begins with the segmentation stage where an adapted version of the *simple linear iterative clustering (SLIC)* algorithm for dealing with hyperspectral data is used. Consequently, the *Hierarchical Agglomerative Clustering (HAC)* and *Fuzzy C-Means (FCM)* algorithms are employed to efficiently group similar superpixels considering distances with respect to each other. The distinct utilization of these clustering techniques defines a complementary stage for overcoming class overlapping during image generation. Ultimately, a *CNN* classifier is trained using the computed image to pixel-wise predict classes on unseen datasets. The labeling results, obtained using two hyperspectral benchmark datasets, indicate that the current approach is able to detect objects boundaries, automatically assign class labels to the entire dataset and to classify new data with a prediction certainty of 90%. Additionally, this method is also capable of achieving better classification accuracy and visual correspondence with reality than the ground truth images.

1. INTRODUCTION

Hyperspectral data consists of a collection of scene radiance arranged in a spatial-spectral datacube. It conveys much more spectral information than the *RGB* color space and other multispectral data, holding pixels as high-dimensional vectors comprising measurements from hundreds of adjacent narrowband channels (Signoroni et al., 2019). This datacube provides the necessary information to discern a wide range of physical phenomena, including mineral type, atmospheric temperature structure, crop health, and cancer cell development (Puschell, 2000). The previous fact determines the multiple applications in scientific and industrial sectors for this kind of data. Biomedicine, food quality, agriculture, and cultural heritage represent only a few of these utilization fields (Signoroni et al., 2019).
Within the last decades, hyperspectral imaging systems have been mainly developed by a range of private enterprises or by government institutions.
The instruments in the first group enable users to collect sensor data anywhere at any time. They are built with advanced technology and own the ability to acquire images with high spatial and spectral resolution. These sensors are easy to handle and can be coupled to an *Unmanned Aerial Vehicle (UAV)* for measuring above small to middle-sized landscapes.
Regarding the government-subsidized missions, they have been first launched into space at the start of the millennium. Since then, state-associated institutions have been planning several hyperspectral imagers. Among the most emblematic European missions is CHRIS, which was the prime instrument of the Proba-1 spacecraft launched on the 22 October 2001. This sensor setup aims to explore the capabilities of imaging spectrometers on agile small satellite platforms (ESA, 2020a). Another representative satellite spectrometer is constituted by the Environmental Mapping and Analysis Program (EnMAP), which strives to monitor and characterize the earth's environment on a global scale. Its start date was last year, and its operational period is 5 years (DLR, 2018). Furthermore, other missions such as *PRISMA* (ESA, 2020b) and the planned *HyspIRI (Hyperspectral Infrared Imager)* (NASA/JPL, 2020) are going to be fully operational within the upcoming years, providing large amounts of reliable and up-to-date data of earth's surface and their interaction with the atmosphere.
In remote sensing, the analysis of hyperspectral imagery constitutes a powerful instrument for solving the task of land cover classification. This relevant activity provides the background information necessary to make informed decisions. The prime knowledge essential to guarantee crop yields represents an illustration of the mentioned required information. The timely detection of water-related stresses increases the chances of a successful crop. These strains are noticeable within variations in photosynthetic pigments leading to yellowish tint in crops, which is efficiently captured by spectrometers due to the increased reflectance of red wavelength (Khan et al., 2018).
Due to the relevance of these decisions, it is beneficial to provide the means to swiftly and automatically generate land cover information instead of waiting for the preparation of the ground truth. These days it is possible to adapt the calculations of the classification process to use *Graphics Processing Units (GPUs)* (Wuttke et al., 2018). This adaptation enables more rapid processing during classification. Hardware resources

should be tightly coupled with performant algorithms to resolve the complexity of the classification problem more efficiently.
In this paper, a method for labeling hyperspectral data to classify unseen datasets through *CNN*s is proposed. The initial stage, the segmentation, loosens the rigid structure of the image to encourage class separability and simplify the upcoming phases. The two-step clustering takes advantage of the class separability in the data to generate a sharp classification. Finally, a *CNN* model is trained to classify new datasets. The capability of this approach is demonstrated for the task of land cover classification on two benchmark datasets.

2. RELATED WORK

With the aim of generating an image of classified land cover, multiple research works have been published within the last few years. The current section describes scientific research closely related to each building block of the proposed method.

2.1 Segmentation

The first stage, the segmentation, describes the process of partitioning an image into segments or groups to simplify the image representation into more meaningful and easier to analyze objects (Shapiro, Stockman, 2001). This step is applied to a hyperspectral datacube. For this task, the work of (Wuttke et al., 2018) was considered due to the fact that it employs an adapted version of the *SLIC* algorithm, where the spectral similarity measured by the Euclidean distance has been effectively replaced by the *Spectral Angle Mapper (SAM)*. The decision of employing a superpixel-based method is attributed to the fact that these are among the most successful segmentation approaches available (Achanta et al., 2012).
In their paper (Achanta et al., 2012), the authors deal with the necessary characteristics, which an algorithm must accomplish to be considered functional. They perform an empirical comparison between the five state-of-the-art superpixel algorithms concentrating on their ability to adhere to image boundaries, speed, memory efficiency, and their impact on segmentation performance. They found out that the *SLIC* outperforms the existing superpixel methods in nearly every respect. The previous fact constitutes the deciding factor for integrating this algorithm into the current approach.

2.2 Clustering

The feature space is an important term to consider when clustering hyperspectral data. It is defined as an imaginary room which limits are determined by the range of the hyperspectral bands (Puschell, 2000). For the task of hyperspectral clustering, which aims grouping objects together that are close in this feature space, several techniques have been successfully applied. One of them is constituted by the *Hierarchical* methods, which have relevancy within a variety of remote sensing applications (Muñoz-Marí et al., 2012). They cluster multidimensional pixels by iteratively grouping them in accordance with a similarity measure. The grouping is executed bottom-up by aggregating pixels (Ward, 1963) or top-down through iterative partitioning (Kashef, Kamel, 2009). The similarity between samples is determined using a similarity function, which most commonly uses the Euclidean or cosine distance (Muñoz-Marí et al., 2012). After having performed the clustering process, a description of the input data can be generated as a hierarchical tree, mostly known as *dendrogram*, which is subsequently studied to determine the required partition level.

Another interesting concept exploited in remote sensing is defined by the *Soft Classification (SC)* techniques. These are characterized by the utilization of soft computing paradigms such as *Fuzzy Logic (FL)*, *Artificial Neural Networks (ANN)*, and genetic algorithms. Their advantage is the provision of more flexibility by exploiting tolerance and uncertainty of real life phenomena (Choodarathnakara et al., 2012). One key method within this group is represented by the *Fuzzy C-Means (FCM)* algorithm. It is characterized by clustering each data point to some degree specified by a membership grade, which models the level of uncertainty or *Fuzzyness* contained within each individual data point (Bezdek, 1981). This technique is particularly relevant when dealing with data points, which are very close to each other in a particular domain (Choodarathnakara et al., 2012) .

2.3 Classification

The final step is the classification of hyperspectral imagery. Most of the currently developed hyperspectral classifiers use *CNN*s because, compared with traditional classification methods, deep-learning-based classifiers have great potential to obtain high classification performance when facing complex inputs (Ghamisi et al., 2018).
An interesting work is presented by (Kumar Roy et al., 2019), where the authors designed a *Hybrid Spectral Convolutional Neural Network (HybridSN)* for hyperspectral image classification. They perceived that most published works were based on *two dimensional (2D) CNN*s even though the classification performance highly depends on both spatial and spectral information. Their algorithm is based on a spatial-spectral *three dimensional (3D) CNN* followed by a spatial *2D CNN*. The *3D CNN* eases the joint spatial-spectral feature representation from a pile of spectral bands while the *2D CNN* learns more abstract representations at spatial level. This configuration allows *hybrid CNNs* to reduce model complexity compared to *3D CNN* alone. Their satisfactory results are compared with the state-of-the-art handcrafted as well as end-to-end deep learning based methods.
Another work using *Deep Learning* applied to hyperspectral analysis is presented by (Hu et al., 2015), where the authors used *deep convolutional neural networks (DCNN)* to directly classify hyperspectral imagery in the spectral domain. Their work resulted in classification outcomes which can perform better than *Support Vector Machine (SVM)* classifiers.

3. METHODOLOGY

This section describes the followed steps during each phase of the proposed method. The approach starts with the segmentation, goes further with the clustering, and lastly, the *CNN* based classification takes place.

3.1 Segmentation

This phase takes the hyperspectral datacube and results in a set of superpixels containing their respective mean spectral signature. This mean signature is calculated considering each individual signature of pixels contained in a superpixel. This phase pursues three purposes: the dissolution of the rigid two-dimensional image structure, the elimination of redundancies and the complexity reduction of the upcoming stages. To achieve these goals, the *SLIC* algorithm has been used (Achanta et al., 2012). Taking as input a desired number of approximately equally-sized superpixels K, it performs a local clustering of pixels in a *five dimensional (5D)* space defined by the L,

a and b values of the *CIELAB (Commission Internationale de l'Eclairage)* color space and the spatial coordinates xy on the image. In order for the algorithm to operate in the 5D space, a distance measure considering superpixel's size is introduced. It enforces color similarity as well as pixel proximity in this 5D room, ensuring that the expected cluster sizes and their spatial extent are approximately equal (Achanta et al., 2010). The distance function together with its components is shown in Equation 3.

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \tag{1}$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{2}$$

$$D = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2} \tag{3}$$

In Equation 3, d_c and d_s are the Euclidean distances of the color and spatial components respectively. The values x_i, y_i and x_j, y_j represent two pair of pixel coordinates located within the search area of the algorithm. The compactness of the resulting superpixels is controlled by the value of m.
The presented color distance measure cannot be meaningfully applied to hyperspectral data because this data contains *RGB* space information and many more channels to be considered. Due to this fact, an adaptation of the color space distance is necessary (Wuttke et al., 2018). This adoption considers the spectral signature of each pixel of the image to compute the *Spectral Angle (SA)* between cluster center and actual pixel. It is written as a function of the scalar product and the multiplication of the L_2-norms of the corresponding spectral vectors. Equation 4 depicts the previous formulation.

$$d_{sa(a,b)} = \arccos\left(\frac{\langle a, b\rangle}{\|a\|_2 \|b\|_2}\right) \tag{4}$$

After obtaining the SA, a replacement in the formulation of the color distance takes place. For the purpose of segmenting spectral pixels, the adapted signature-based distance is expressed in Equation 5.

$$D' = \sqrt{\left(\frac{d_{sa}}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2} \tag{5}$$

Additionally to this formulation, the value of the used parameter m has been treated as an adjustable constant which ensures regular superpixels form and, most importantly, a high capability of the superpixels to adhere to image boundaries (Achanta et al., 2012).
After performing the segmentation, a supervised validation process takes place. This validation is realized by a visual evaluation of the generated superpixels considering their form, and how well they adhere to pixel boundaries.

3.2 Clustering

The second phase deals with the methods utilized to group similar superpixels, and assign them labels. The upcoming sections describe in detail the two employed techniques.

3.2.1 Hierarchical At first, the *Hierarchical Agglomerative Clustering (HAC)* in its *bottom-up* version is used. This method has been selected over the originally proposed *bisecting k-means (BKM)* because it does not need the apriori definition of the desired number of clusters and it works pretty well on remote sensing data (Muñoz-Marí et al., 2012). Additionally, it possesses fast runtimes for small up to medium-sized datasets and its implementation is clear to follow. It begins by creating a clustering multilevel hierarchy out of the superpixels, where clusters at one level are part of clusters defined at the next level (MathWorks, 2020b). This agglomeration occurs in the feature space. For the correct functioning of this method, the three following phases are recognized:

Similarity Measurement This step is in charge of finding the similarity between every pair of superpixels in the dataset. The distances between superpixels are computed considering three different metrics. First, *SAM* is utilized whose formulation is based on Equation 4. Afterwards, the *Euclidean* and the *cosine distance* metrics were evaluated. They have been selected because they represent frequently used metrics for the classification of multi- and hyperspectral data (Wuttke, 2018). Their respective formulations can be seen in Equations 6 and 7.

$$d_{eu(a,b)} = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{6}$$

$$d_{cos(a,b)} = 1 - \frac{a \cdot b}{\sqrt{(a \cdot a)(b \cdot b)}} \tag{7}$$

In Equation 6, n represents the number of hyperspectral bands.

Linkage This phase groups the superpixels into a binary hierarchical cluster tree. Pairs of superpixels that are in close proximity are linked using the distance information created during *Similarity Measurement*. As superpixels are paired into binary clusters, the newly formed clusters are grouped into larger ones until the hierarchical tree is completed (MathWorks, 2020b). Two linkage functions have been used in combination with the generated similarity. The first is the *average* linkage in which, for each pair of clusters, the distance of two clusters is calculated as the average of the distances of each element of the cluster with each element of the other cluster. Then, it merges the clusters together to minimize the maximum distance between the clusters (Manning et al., 2008). The second function is the *ward* linkage in which for each cluster an error function is defined. This error function is the average *Root Mean Squared (RMS)* distance of each data point in a cluster to cluster's center of gravity (Ward, 1963).
During this process, the *SAM* distance was used together with the *average* linkage. Moreover, the *Euclidean* distance used *ward*, and the *Cosine* also employed the *average* function.

Groupping This step determines where exactly the hierarchical tree will be cut to generate clusters. The branches at the bottom of the hierarchy have been pruned off and all the superpixels below each cut has been assigned to a single cluster. The branching off has been executed visualizing the marked divisions in the data by using a *dendrogram* diagram which will be presented in Chapter 6. The respective clusters boundaries have been found by observing the vertical separations between each binary level of the hierarchical tree.

At this point, the validation of the *HAC* result is executed. This process was done by visually comparing the *RGB* image with the corresponding generated one. The corresponding ground truth images did not offer a comparison frame for a direct association. They possess too few labeled samples, and the labeled forms do not correspond with reality. Additionally, in the case of the second dataset, the definition of the classes *asphalt*, *gravel*, *bitumen*, and *bricks* are too application-specific, and their crisp separation using clustering is a challenging topic of further investigation.

3.2.2 Fuzzy The phenomenon of class overlapping is widespread in practice in different engineering fields, and it represents one of the toughest problems to solve in classification. It happens when objects belonging to different classes possess very similar characteristics. These elements reside in overlapping regions within the feature space and are often localized near to class boundaries, which leads to misclassification (Xiong et al., 2010).
This hurdle represents a well-known challenge when dealing with remote sensing data. In the present study, both used datasets experienced this effect, leading the team to find the best-fitting answer to solve this problem. The familiy of the *Fuzzy Clustering* algorithms has been chosen to deal with this hurdle. This category was considered because it represents an effective technique used in remote sensing which incorporates collateral data easily so that similar land cover can be well classified (Choodarathnakara et al., 2012). The used algorithm is *FCM*, which groups data points distributed in a multidimensional space into a specific number of clusters (MathWorks, 2020a).
After applying this method, a validation process was executed. This validation was realized by a visual inspection considering a crisp partition of the seeked classes.

3.3 Classification

The current section describes the classification phase involving the training, validation and prediction tasks of a *CNN* model. This stage aims the semantic classification of unseen pixels of a hyperspectral datacube. For this purpose, an initial *CNN* architecture was realized considering the method developed by (Hu et al., 2015). The fact that led this study to look upon the cited work was the idea to treat each spectral pixel as an image. In their work, the authors remark that each hyperspectral pixel sample can be handled as a *2D* image whose height is equal to 1, as audio inputs are treated in speech recognition. Therefore, the size of the input layer is $1 \times n_d$, where n_d represents the number of bands. The previous leads to the concept of *1D CNN* (Hu et al., 2015).
The initial *CNN* architecture has been progresively optimized considering an iterative adaptation process. During this period, it is aimed to keep the model as simple as possible. Several hyperparameters of the model were slightly variated and tested, executing test runs to evaluate the impact of these changes on network performance. Furthermore, the developed model was continually controlled against overfitting, which was determined by the presence of a validation set during training. The final developed *CNN* architecture is shown in Figure 1.
In Figure 1, the parameters $n1$ to $n6$ represent the output shape at each layer and they are specific to each dataset. Their respective values are presented in Chaper 6.
The upcoming training section reveals the hyperparameters tuned during this phase. Subsequently, the validation section describes the employed techniques to ensure verification during and after training. Finally, the prediction section describes the procedure to predict each spectral pixel from the unseen datacube.

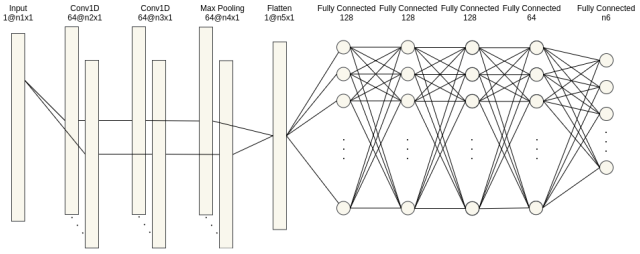


Figure 1. Final *CNN* architecture for both datasets.

3.3.1 Training The parameters used during model training were the *Categorical Crossentropy* loss function, the *Adam* optimizer, a batch size of 128, 30 to 100 epochs, and a learning rate of 0.001.

3.3.2 Validation The validation has been executed both during and after the training phase. For this purpose, a validation and a test split were respectively prepared using the input datacube.

3.3.3 Prediction This phase involves the class prediction of each unseen spectral pixel. To define a quality assessment, a difference image between the labels and the predictions was computed. This calculation considers the approach developed by (Wang et al., 2004), which is based on the degradation of structural image information. This method is based on the *Structural Similarity (SSIM)*, which is a decimal value between $[-1, 1]$ used to measure the similarity between two images. This allows to perceive the discrepancies between each map, delineating possible deficiencies of the *CNN* model.

4. DATASETS

This section describes the characteristics of each used dataset.

4.1 Greding

The first benchmark dataset is known as *Greding* and was acquired by researchers at the Fraunhofer IOSB in July 2014. It depicts a portion of the central region of the Greding village in the southwest of Germany. It was collected using an *aisaEAGLE II* airborne sensor. The dataset is composed of 127 bands covering the electromagnetic spectrum from 390 to 990 nanometers. Its extension is of 670×606 pixels, with a spatial resolution of 0.5 meters. The ground truth contains six classes with 127 688 labeled samples (Gross et al., 2019).

4.2 Pavia University

The second data collection is the *Pavia University* scene, acquired by the German Aerospace Centre (DLR) within the scope of the HySens project (Mater, 2014). Its geographical target was the Engineering School of the University of Pavia in Italy. The used sensor was the *ROSIS-03*, creating a datacube composed of 103 spectral bands covering spectrum values from 430 to 860 nanometers. The image has an extension of 640×340 pixels, in which each of them has a spatial resolution of 1.3 meters. The ground truth data consists of nine classes with a total of 42 776 labeled samples (Graña et al., 2020).
The selection of these sets was done considering a suitable class separability and the respective observation of landscape scenes.

5. EXPERIMENTAL SETUP

This section explains the experimental runs and the resources used during each step. The runs are based on each phase described in Chapter 3. The run configurations are described as it follows:

1. Segmentation of the *Greding* datacube with the consecutive application of the clustering techniques explained in Chapter 3.2.
2. Using the generated labeled image, the *CNN* model trains, and generates a classification map out of the prediction of unseen data.
3. Making use of the labeled and predicted maps, the difference image is computed.
4. The same procedure is executed with the *Pavia University* datacube.

For the implementation of the segmentation stage, the *Python* programming language, in its version 3.6.9, has been used. The clustering phase uses the *MATLAB* R2019b and R2020a implementations of the *HAC* and *FCM* algorithms respectively. The machine learning framework *TensorFlow* (Abadi et al., 2015) is used for the *CNN* based classification step. Ultimately, the *scikit-image* (van der Walt et al., 2014) implementation of the *SSIM* based algorithm is used to calculate the divergence image. The mentioned software resources were used with an Intel Core i7-6700 processor and 32-GB RAM.

6. RESULTS AND DISCUSSION

In this section, the description of the outcomes obtained by applying the current approach is presented. Each upcoming section describes the corresponding stage in detail.

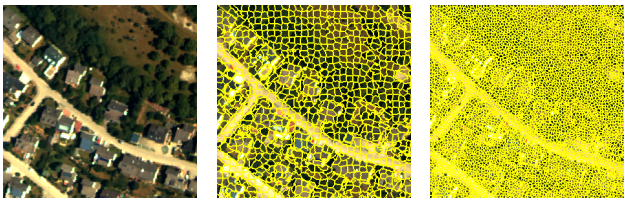
6.0.1 Segmentation First, the results obtained for *Greding* are described. Subsequently, the description of the segmentation results for the second dataset takes place.

Greding As a preprocessing action, the dataset have been normalized in order for the observations to ensure more efficiency during computation. The employed *minimum-maximum* normalization is formulated in Equation 8.

$$X' = \frac{X - min(X)}{max(X) - min(X)} \tag{8}$$

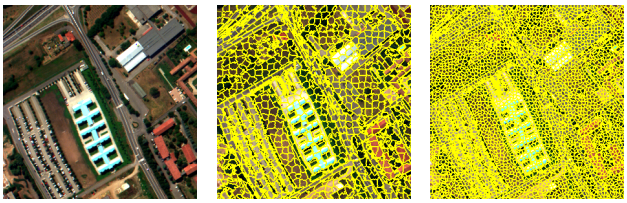
In Equation 8, X' represents the original dataset after normalization (Grus, 2015). For this dataset, approximately 4% of its total pixels number was selected as the desired number of superpixels K. This represents an amount of 16 214 superpixels which provide an optimal oversegmentation level for the upcoming clustering step. The used primary distance metric was *SAM* and the compactness parameter m was determined to be 0.175. A visualization of the segmentation results using two values of K is seen in Figure 2.

Pavia University For this dataset, the parameter K was set to 8 296 which also represents 4% of its number of pixels. This value also defined the most adequate oversegmentation quote. The primary distance metric was also *SAM* and $m = 0.175$. The segmentation results can be perceived in Figure 3.



(a) (b) (c)

Figure 2. Visualization of different segmentation levels for *Greding*. (a) shows a section of the *RGB* image, (b) depicts the result with 1% and (c) with 4% of the total number of image pixels.



(a) (b) (c)

Figure 3. Visualization of the *Pavia University* segmentation. (a) depicts a *RGB* image section, (b) illustrates the segmentation result with 1% and (c) with 4% of the total number of pixels.

6.0.2 Clustering During this phase, the described clustering techniques are extensively used to automatically generate a labeled image. First, the results obtained for *Greding* are described.

Greding The clustering step begins with Section 3.2.1. First, the 16 214 superpixels, together with their respective mean spectrum, are taken as input for the linkage computation. At this step, the *Euclidean* distance metric is used because it resulted in being the best performing metric after numerous experimental runs. At the same time, the used linkage function was *ward*.
After distance computation, a *dendrogram* was generated, and analyzed to determine the optimal value for sectioning the hierarchical tree. A value of 5.605 was used to divide the mentioned tree in 6 parts, which corresponds to the total number of targeted clusters. A visualization after sectioning is shown in Figure 4.

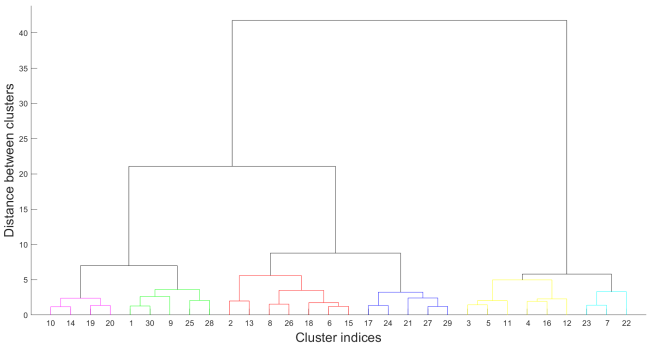


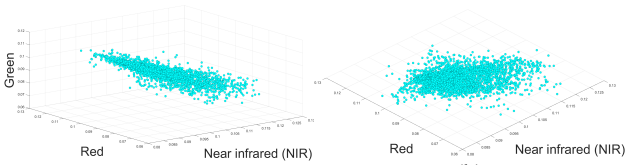
Figure 4. Grouped links during hierarchical tree analysis.

After clusters division at *dendrogram* level, the next step was to used the mentioned sectioning value to generate the clusters. A visualization of the preliminary created classes is seen in Figure 5.
In Figure 5 (b), the class buildings (1 on the colorbar) is composed of well-distinguished buildings and shadows. To be in accordance with the number of classes contained in the ground

(a) (b)

Figure 5. Preliminary labeled image for *Greding* after hierarchical tree division.

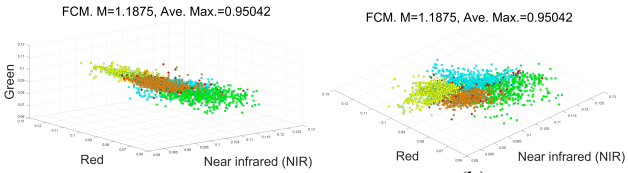
truth, the natural separation of these two elements is not considered and represents a topic of further investigation. At this point, the algorithm could not identify buildings still contained in the streets class (4 on the colorbar). This error is accredited to the fact that streets and buildings represent high reflectant bodies and the recorded data confirms very similar radiance values. This problem has been described in Section 3.2.2 and it is resolved employing the *FCM* algorithm.
To observe the overlapping level of the buildings and streets data, Figure 6 shows the superpixels distribution in feature space.



(a) (b)

Figure 6. Buildings-streets superpixels in feature space.

Figure 6 shows that a clear separation of the superpixels cannot be distinguished and they are pretty close to each other. To compute the *FCM* clustering, different overlapping levels were tested to determine the value ensuring a distinct class separation. The studied overlap range was $[1.10 - 1.45]$, where 1.187 was the selected optimal value. This value was chosen after an experimental iterative process and it is used to classify each superpixel. Another important parameter to consider is the number of desired clusters. It can be inferred that the input number of clusters would be 2 (for buildings and streets) but a number of 4 clusters was required. This quantity yielded the best results while separating both classes. The results of the *fuzzy* classification is shown in Figure 7.

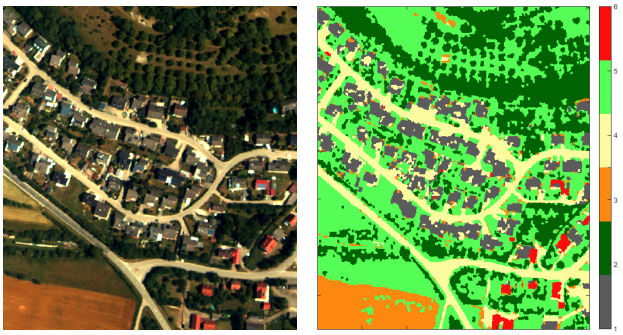


(a) (b)

Figure 7. Buildings-streets superpixels after classification.

Figure 7 shows a representation of the clustered superpixels where each red point represents an ambiguous or *fuzzy* superpixel. The average maximum membership value (Ave. Max.) provides a quantitative description of the overlap. The value 0.950 indicates crisp clusters with low overlap level.

The classes were, in a further processing step, merged together considering only the superpixels that corresponded to buildings and streets depicted on the original *RGB* image. It can be said that this merging phase was executed in a supervised manner. The next step was the combination of the *HAC* and *FCM* results. The generated labeled image is visualized in Figure 8.



(a) (b)

Figure 8. *RGB* and labeled image visualization. (a) shows the *Greding* scene and (b) the automatically generated image.

In Figure 8, note how some pixels are still identified as streets even though they in reality belong to the buildings class. This is attributed to the fact that these pixels are part of superpixels with maximum membership values below 0.6. This means that they experienced a more *fuzzy* classification and can either belong to the buildings class or to the streets one. This *fuzzyness* in the data points reveals how similar they are in the feature space so that they can adopt 2 different states.

Pavia University For this dataset, the clustering step follows the same sequence described with *Greding*. First, the linkage computation with $8\,296$ superpixels was executed. The *Euclidean* distance metric as well as the *ward* algorithm were also employed. The *dendrogram* inspection resulted in a value of 5.2 as the distance to section the hierarchical tree into 7 parts. Figure 9 shows a visualization of the grouped links.

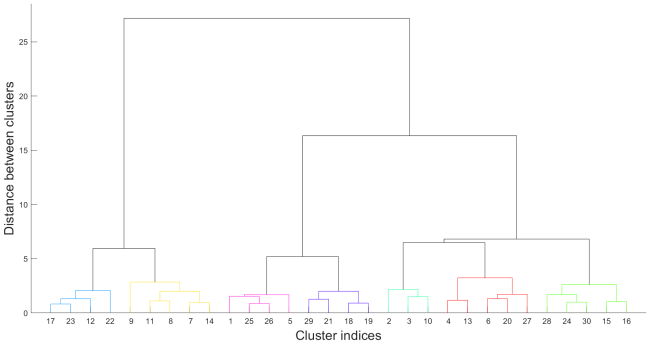
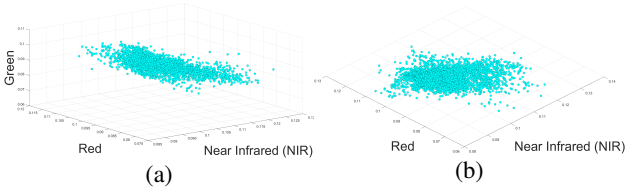


Figure 9. Grouped links depicting the number of clusters.

The next step used the separation value for clusters generation. An illustration of the preliminary created clusters is seen in Figure 10.
As seen in Figure 10 (b), 7 classes were generated because this amount represents an optimal level for clustering most of the classes. It is also seen that there is no distinction between streets and buildings. This challenge was again solved using *FCM*. The unprocessed superpixels are shown in Figure 11.
Figure 11 evidences that there are no built groups that can be properly identified. To find this separation in the data, different overlapping levels were also evaluated. The studied range was

(a) (b)

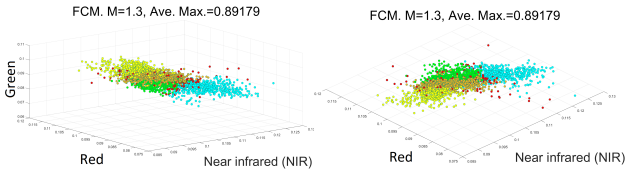
Figure 10. Preliminary labeled image for *Pavia University* after hierarchical tree partition.



(a) (b)

Figure 11. Buildings-streets superpixels in feature space.

$[1.10 - 1.5]$, selecting an overlapping value of 1.3. Similarly to *Greding*, a number of 4 desired clusters was required to perform the classification. The results are shown in Figure 12.



(a) (b)

Figure 12. Buildings-streets superpixels after classification.

As seen in Figure 12, an average maximum membership value of 0.891 indicates low overlap.
After merging corresponding superpixels together, the combination of the two clustering results took place. The labeled image is visualized in Figure 13.

6.0.3 Classification Finally, the outcome of the previous phase is utilized to train the *CNN* model for predicting labels. The description of this step also follows the datasets sequence.

Greding The *Greding* datacube has been taken to generate the data splits for the learning phase. Three subsets have been prepared, defined by the training, validation and test splits. Their data proportion was 50%, 25% and 25% respectively.
After preprocessing, the definition of the n model parameters takes place. These values were determined after building the model, and using the summary *TensorFlow* functionality. The values for this dataset are: $n1 = 127$, $n2 = 114$, $n3 = 101$, $n4 = 33$, $n5 = 2\,112$, and $n6 = 6$.

(a) (b)

Figure 13. Resulting labeled image. (a) shows the *Pavia University* scene and (b) the generated labeled image.

During training, 100 epochs have been sufficient in order for the model to learn relevant features for the classification. During each epoch, the model used the validation split to continuously evaluate the loss and the accuracy as primary model metrics. After having trained, the model learned with an overall accuracy of 90%, an adequate accuracy level for the goals of the current classification. Additionally, a final divergence factor of 0.011, between training and validation accuracy, determines a moderate overfitting level.
The processing pipeline continues with the model evaluation. For that, the test split was used. The result is the model having a test loss and accuracy of 0.141 and 0.893 respectively. These obtained metrics evidence how well the model performs on unseen data.
Next, the prediction of each pixel contained in the mentioned test set was executed. For each unseen pixel, the model outputed a mean prediction certainty of 90% that each pixel belongs to a certain class. To quantify the divergence level between the labeled and the predicted map, the difference image was computed and it is shown in Figure 14. The *SSIM* value was also calculated during this process. A value of 0.982 confirms that there is no significant difference between the images and that they are close to the perfect match represented by a value of 1.0.
Figure 14 depicts the distribution of each compared pixel around the scene. The colorbar shows both the regions experiencing more discrepancy (intensive red) and the areas having no difference at all (intensive blue). To rely on a quantitative measure of quality, a *SSIM* threshold of 0.9 has been defined. The quantity of pixels laying under this threshold is 10 465 and the rest (395 555) represents pixels with larger *SSIM*s. This means that more than 95% of the pixels have been correctly classified.

Pavia University Similarly to *Greding*, three splits with the same data proportion have been created.
The model's output parameters possessed the following values: $n1 = 103$, $n2 = 90$, $n3 = 77$, $n4 = 25$, $n5 = 1\,600$, and $n6 = 9$.
The training phase required 30 epochs for the classifier to learn. The model learned with an overall accuracy of 82%, a suitable level for the goals of the classification. A final divergence factor

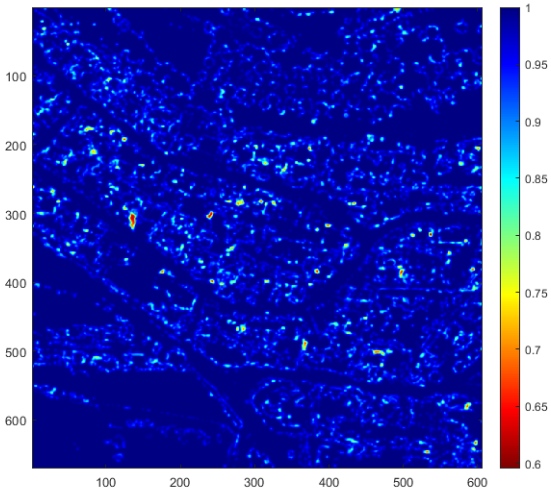


Figure 14. *Greding* difference image.

of 6.3^{-3}, between training and validation accuracy, determines, to a certain extent, a low overfitting level.
The next step is the model evaluation. The obtained test loss and accuracy are 0.485 and 0.807 respectively. As these metrics evidence, the model do perform well on the test dataset.
During the prediction of each new pixel, the model outcomes a 82% of certainty that each pixel belongs to its corresponding class.
The difference image and the *SSIM* value were correspondingly computed. The image is presented in Figure 15. The calculated *SSIM* of 0.973 confirms that there are no major differences between the compared images.

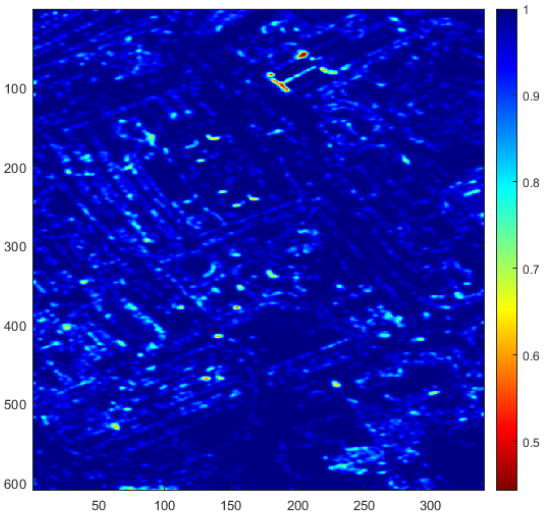


Figure 15. *Pavia University* difference image.

Figure 15 shows the distribution of each pixel around the scene. The quantitative quality measure is also defined by the threshold of 0.9. The amount of pixels laying under this limit is $10\,539$, and the rest $(196\,861)$ have larger *SSIM*s. The previous evidences that more than 94% of the pixels have been correctly classified.

7. CONCLUSION

The main goal of the presented study was the creation of an unsupervised method to automatically label hyperspectral pixels with the aim of using them as labels for *CNN* classification. The method combines well-established computer vision techniques such as segmentation, clustering and deep learning to materialize a processing pipeline for land cover classification. The segmentation stage outcomed an adequate number of spectral superpixels which was crucial for the computation performance of the upcoming steps. The *HAC* step could properly infer the hierarchy contained within the superpixels and outputted well-separated land cover classes where boundaries were unambiguous. Within this clustering frame, *FCM* was in charge of making the distinction between very similar classes living in the feature space. It outputted satisfactory classification results considering the well-known difficulty of separating overlapping classes. At this point, it is indispensabel to further investigate this problem making use of alternative approaches such as *ANN*s (Choodarathnakara et al., 2012). At the final stage, the *CNN* was in charge of learning hyperspectral features to be reusable and able to classify unseen datasets with a high level of accuracy.
In conclusion, the current approach successfully combined three different machine learning techniques into one algorithmic processing chain to efficiently automate the computation of a land cover classification. This approach removed the labelling costs, investing the gained time in activities requiring valuable human intervention such as class overlapping problem solving or *CNN* optimization. This method was also well-suited to recognize similarity patterns between objects better than the human eye could do. Finally, its modular nature and flexibility provides the chance of integrating other methods into it, adapting each stage to specific needs. An illustration would be the usage of *ANN*s within the clustering stage without the need of modifying the other phases. Similarly, in the classification stage, the classifier could be easily changed by a different algorithm, for instance, by a *SVM*. Cost reduction, proficiency, and versatility are the traits of the presented study.

REFERENCES

Abadi, M. et al., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2010. SLIC Superpixels. *EPFL Technical*, Report 149300.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2012. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274-2282.

Bezdek, J., 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press.

Choodarathnakara, A., Ashok, T., Shivaprakash, K., Patil, C., 2012. Soft Classification Techniques for RS Data. *IJCSET*, 2(11), 1468-1471.

DLR, 2018. Enmap. https://www.enmap.org/.

ESA, 2020a. Chris. https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/proba/instruments/chris.

ESA, 2020b. Prisma (hyperspectral precursor and application mission). https://directory.eoportal.org/web/eoportal/satellite-missions/.

Ghamisi, P., Maggiori, E., Li, S., Souza, R., Tarabalka, Y., Moser, G., De Giorgi, A., Fang, L., Chen, Y., Chi, M., Serpico, S., Benediktsson, J., 2018. New Frontiers in Spectral-Spatial Hyperspectral Image Classification. *IEEE Geoscience and Remote Sensing Magazine*.

Graña, M., Veganzons, M., Ayerdi, B., 2020. Hyperspectral remote sensing scenes. http://www.ehu.eus/ccwintco/index.php.

Gross, W., Tuia, D., Soergel, U., Middelmann, W., 2019. Nonlinear feature normalization for hyperspectral domain adaptation and mitigation of nonlinear effects. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8), 5975–5990.

Grus, J., 2015. *Data Science from Scratch.* O'Reilly.

Hu, W., Huang, Y., Wei, L., Zhang, F., Li, H., 2015. Deep Convolutional Neural Networks for Hyperspectral Image Classification. *Journal of Sensors*, 2015(258619), 12.

Kashef, R., Kamel, M., 2009. Enhanced bisecting k-means clustering using intermediate cooperation. *Pattern Recognition*, 42(11), 22572569.

Khan, M., Khan, H., Yousaf, A., Khurshid, K., Abbas, A., 2018. Modern Trends in HyperspectralImage Analysis: A Review. *IEEE Access*.

Kumar Roy, S., Krishna, G., Dubey, S., Chaudhuri, B., 2019. HybridSN: Exploring 3D-2D CNN FeatureHierarchy for Hyperspectral Image Classification. *IEEE Geoscience and Remote Sensing Letters*.

Manning, C., Raghavan, P., Schütze, H., 2008. *Introduction to Information Retrieval*. Cambridge University Press. Chapter 17. Hierarchical clustering.

Mater, J., 2014. The hysens project. https://www.hysens.eu/.

MathWorks, 2020a. Fuzzy clustering. https://de.mathworks.com/help/fuzzy/fuzzy-clustering.html.

MathWorks, 2020b. Introduction to hierarchical clustering. https://de.mathworks.com/help/stats/hierarchical-clustering.html.

Muñoz-Marí, J., Tuia, D., Camps-Valls, G., 2012. Semisupervised Classification of Remote Sensing Images With Active Queries. *IEEE Transactions on Geoscience and Remote Sensing*, 50(10).

NASA/JPL, 2020. Hyspiri mission study. https://hyspiri.jpl.nasa.gov/.

Puschell, J., 2000. Hyperspectral imagers for current and future missions. *Proceedings SPIE 4041, Visual Information Processing IX*.

Shapiro, L., Stockman, G., 2001. *Computer Vision*. Prentice Hall.

Signoroni, A., Savardi, M., Baronio, A., Benini, S., 2019. Deep Learning Meets Hyperspectral Image Analysis: A Multidisciplinary Review. *Journal of Imaging*.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., the scikit-image contributors, 2014. scikit-image: image processing in Python. *PeerJ*, 2, e453. https://doi.org/10.7717/peerj.453.

Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*.

Ward, J., 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301), 236244.

Wuttke, S., 2018. Aktives Lernen mit Segmentierung und Clusterbildung zur bildbasierten Klassifikation der Landbedeckung. PhD thesis, Technische Universität München.

Wuttke, S., Middelmann, W., Stilla, U., 2018. Improving the Efficiency of Land Cover Classification by Combining Segmentation, Hierarchical Clustering, and Active Learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Xiong, H., Wu, J., Liu, L., 2010. Classification with Class Overlapping: A Systematic Study. *International Conference on E-Business Intelligence*.