remote sensing

MDPI

Article

Improving Mineral Classification Using Multimodal Hyperspectral Point Cloud Data and Multi-Stream Neural Network

Aldino Rizaldy *, Ahmed Jamal Afifi, Pedram Ghamisi and Richard Gloaguen

Helmholtz-Zentrum Dresden-Rossendorf (HZDR), Helmholtz Institute Freiberg for Resource Technology (HIF), 09599 Freiberg, Germany; a.afifi@hzdr.de (A.J.A.); p.ghamisi@hzdr.de (P.G.); r.gloaguen@hzdr.de (R.G.)
* Correspondence: a.rizaldy@hzdr.de

Abstract: In this paper, we leverage multimodal data to classify minerals using a multi-stream neural network. In a previous study on the Tinto dataset, which consisted of a 3D hyperspectral point cloud from the open-pit mine Corta Atalaya in Spain, we successfully identified mineral classes by employing various deep learning models. However, this prior work solely relied on hyperspectral data as input for the deep learning models. In this study, we aim to enhance accuracy by incorporating multimodal data, which includes hyperspectral images, RGB images, and a 3D point cloud. To achieve this, we have adopted a graph-based neural network, known for its efficiency in aggregating local information, based on our past observations where it consistently performed well across different hyperspectral sensors. Subsequently, we constructed a multi-stream neural network tailored to handle multimodality. Additionally, we employed a channel attention module on the hyperspectral stream to fully exploit the spectral information within the hyperspectral data. Through the integration of multimodal data and a multi-stream neural network, we achieved a notable improvement in mineral classification accuracy: 19.2%, 4.4%, and 5.6% on the LWIR, SWIR, and VNIR datasets, respectively.

Keywords: mineral classification; geology; multimodal; hyperspectral; point cloud; deep learning; graph-CNN; hyperclouds; data fusion; multi-stream

Citation: Rizaldy, A.; Afifi, A.J.; Ghamisi, P.; Gloaguen, R. Improving Mineral Classification Using Multimodal Hyperspectral Point Cloud Data and Multi-Stream Neural Network. *Remote Sens.* **2024**, *16*, 2336. https://doi.org/10.3390/rs16132336

Academic Editors: Yanni Dong, Tao Chen and Chao Chen

Received: 8 May 2024
Revised: 20 June 2024
Accepted: 24 June 2024
Published: 26 June 2024

1. Introduction

Several studies [1–5] have focused on identifying mineral classes using hyperspectral data. Traditionally, these studies treated hyperspectral data as 2D images, with the spectral information represented by the depth of the image. However, in tasks such as mineral classification, particularly in challenging environments like open-pit mines, representing the real-world situation accurately is difficult using 2D images alone. Instead, 3D data, such as point clouds, offer a more comprehensive representation of the complex environment. Recent research [6–8] has integrated hyperspectral data with point clouds, resulting in what is termed 'hyperclouds', for various applications in mineral exploration and mining monitoring. Similarly, our previous work [9] integrated the hyperspectral data of multiple sensors from different views with 3D point clouds, creating a multimodal dataset comprising three distinct modalities: XYZ coordinates, hyperspectral data, and RGB images.

While we achieved success in identifying mineral classes using various deep learning models for point clouds, our prior work [9] only utilized hyperspectral data as input, overlooking the potential benefits of exploiting the multiple modalities present in our dataset. In this study, we aim to investigate the prospect of enhancing mineral classification accuracy by leveraging the multimodal nature of the dataset.

Training a multimodal dataset is often approached as a data fusion problem in machine learning. Fusion models typically fall into categories such as early, late, and intermediate

fusion [10–12]. In our investigation, we explore various fusion architecture designs to develop efficient and reliable models for our task. Our experiments indicate that intermediate fusion yields the best performance, highlighting the benefits of utilizing a multi-stream network for multimodal data.

Our network architecture consists of two main parts. The first part comprises multiple branches, which are each dedicated to a different modality within the dataset. In the second part, we incorporate a single-stream network that aggregates the learned features from the previous branches. This design facilitates the effective integration of information from diverse modalities, thereby enhancing the overall performance of the model.

With the advancements in deep learning models for point cloud classification, we have a wide range of architecture options to consider as the backbone of our framework. Many of these backbones rely on local feature aggregation techniques exemplified by models such as PointNet++ [13], DGCNN [14], and Point Cloud Transformer [15].

Our previous work [9] indicated that DGCNN consistently performs well across various hyperspectral datasets. The primary feature of DGCNN lies in its use of the EdgeConv operator, which is known for being lightweight and efficient. Therefore, we opted to employ the EdgeConv operator as the feature encoder in our multi-stream network.

Additionally, as evidenced in various studies where leveraging channel information proves beneficial [16–19], we integrate a channel attention module to exploit spectral information. Our implementation of the channel attention mechanism involves incorporating 3D CNNs on top of the EdgeConv operator within the hyperspectral stream of the network. This implementation draws inspiration from the Efficient Channel Attention Network (ECANet) [19], which employs 1D CNNs as gating mechanisms to modulate channel-wise features. This integration results in notable accuracy improvements.

Finally, we conduct a comparison between point-based classification and image-based classification for the same task. While our dataset primarily consists of 3D hyperclouds, it also includes 2D images as their direct counterpart. Notably, the hyperclouds were derived from backprojected hyperspectral images to 3D point clouds. To ensure equitable comparison with our 3D approach, we adapted the EdgeConv operator for image data. Our experimental results reveal a significantly superior accuracy in point-based classification compared to image-based classification.

The key contributions of our works are as follows.

- We demonstrate that leveraging multimodal data yields superior classification results compared to utilizing unimodal data for mineral classification tasks.
- Our study reveals the network's capability to learn various data modalities present in 3D hyperclouds through the expansion of the network into a multi-stream architecture.
- We introduce an approach of employing 3D CNNs on top of the EdgeConv operator to capture spectral patterns, leading to enhanced segmentation performance.
- We perform a direct comparison of segmentation results between point cloud and image data formats, highlighting the superior performance of point-based segmentation.

The rest of this paper follows the following structure: Section 2 discusses existing works on hyperspectral and point cloud data segmentation. Next, in Section 3, we introduce our hypercloud data used in this work along with the proposed framework. Section 4 presents the results of mineral classification on our dataset, including a comparison between a point-based segmentation and an image-based segmentation. In Section 5, we conduct ablation studies to further understand the significance of each component of our framework. Finally, Section 6 provides the conclusion of our work.

2. Related Works

2.1. Hyperspectral Data Segmentation

For a long time, hyperspectral features have been used as predictors for various image classification tasks, including remote sensing data for land cover classification [20,21]. Unlike standard three-channel RGB images, hyperspectral data often consist of hundreds of spectral channels. In the past, the exploitation of spectral features as the only

predictors [22–26] has been the common way to classify pixels of hyperspectral data. Having identified the predictors, any machine learning models can be trained in supervised learning. Although it might result in good classification scores as was reported in some works [27,28], the spatial relationships are omitted. Given the fact that the semantic information of pixels is usually closely related to their neighbors, neglecting the spatial relationship of the pixels can be seen as wasting the full potential of the data. The spatial relationship should be exploited to achieve higher classification scores.

Some unsupervised learning segmentation techniques have been developed to benefit from the spatial relationships [29–31]. The segmentation is usually performed under the assumption that adjacent pixels in the metric domain have a higher probability of being in the same class. The classification can be followed by supervised learning or remains purely unsupervised.

The emergence of CNNs for natural images motivated the community to adopt 2D convolutional filters for the hyperspectral data [32,33]. Furthermore, instead of only using 2D CNNs, some studies [2,3,34,35] pair 2D CNNs with 1D CNNs. The former captures local spatial features, and the latter learns spectral features. However, this approach is not direct and therefore has been avoided lately.

Recent studies [1,4] suggest employing 3D CNN to learn spatial and spectral features simultaneously. It is inevitable since hyperspectral data are commonly presented as 3D cubes rather than 2D rasters. The 3D convolutional filters learn features naturally on the 3D cubes. Hence, the network is simpler and easier to train [4,36].

2.2. Point Cloud Segmentation

In recent years, there has been a significant increase in the popularity of deep learning networks designed for point cloud classification, as noted by researchers such as Guo et al. [37], Bello et al. [38], Zhang et al. [39], and Xie et al. [40]. The 3D point cloud format has become particularly attractive for deep learning applications, as it provides the possibility to extract more comprehensive features such as local and global 3D geometrical features.

Learning point features is quite different from learning features in images because point clouds are unordered, unstructured, and lack a grid format [37–39]. Nevertheless, various approaches have been developed for point cloud classification and segmentation. According to [37], some of these approaches can be categorized as follows:

- Multi-view images [41,42], which project points onto images and use typical 2D convolution networks.
- Occupied voxels with 3D convolution networks, either using dense convolution [43,44] or sparse convolution [45,46].
- MLP-based architectures [13,47] that rely on the max-pooling layer to achieve permutation invariance.
- Graph-based architectures [14,48] that generate k-NN graphs and pass them to MLP-based networks.
- Convolutional-based architectures [49–52] that develop custom point-convolution operators.
- Transformer-based architectures [15,53] that solely use self-attention modules as the encoder.

2.3. Hyperspectral and Point Cloud Fusion

The fusion of hyperspectral data and point clouds has garnered increased attention in recent years. Some studies [54,55] have developed methods for integrating LiDAR and hyperspectral data. The MUUFL Gulfport dataset [56] not only provides an integrated product but also includes corresponding ground truth labels. Other research [57,58] utilizing this dataset has concluded that combining hyperspectral and LiDAR data yields the highest accuracy compared to the data alone. Spectral features enable the classifier to detect the materials of objects, while geometric features are suitable for distinguishing different

object shapes. Additional studies [55,59–61] consistently showed that fusion leads to more promising results compared to working with each data modality separately.

Recent works [62,63] describe the application of deep learning models for the segmentation of hyperspectral point clouds using the urban scene GRSS18 dataset [64]. These models are based on various pixel and point convolution architectures. Surprisingly, an all-pixel convolution with the integration of hyperspectral and LiDAR data outperformed 2D and 3D combinations [62]. Another work [65] also demonstrated the performance of a deep learning model with a different architecture using random point sampling. These studies have motivated the exploration of deep learning and the development of specialized networks for multimodal data.

2.4. Fusion Models for Multimodal Data

In machine learning, fusion models for multimodal data are typically implemented as early fusion, late fusion, or intermediate fusion [10–12]. Early fusion, or data-level fusion, is the data merging from different sensors as input for the machine learning model. Late fusion, also known as decision-level fusion, involves using independent models for each modality and then combining their decisions to improve results. These fusion techniques are considered traditional and were commonly applied before the deep learning era, using classic machine learning models like Random Forest or Support Vector Machine.

Deep neural networks introduce intermediate fusion (feature-level fusion) [10,11]. The flexibility of neural network architecture allows for various fusion schemes, enabling models to fuse learned features at different levels of abstraction. This flexibility also allows different modalities to communicate with each other at different levels, as seen in recent works [66,67]. Additionally, research by [11] showed that intermediate fusion often resulted in higher classification scores than early or late fusion. In [68], different fusion strategies, including dense and sparse fusion approaches, were investigated.

Multimodal remote sensing data have also been widely explored within deep learning frameworks [69]. To improve classification results, some studies [70,71] have fused radiometric and geometric features of remote sensing images. Various early and late fusion networks (e.g., ResUNet, V-FuseNet, SegNet-RC) were briefly studied in [72,73]. The work in [12] goes beyond early, late, or intermediate fusion by extracting features in a cross-modal multi-scale manner. PerceiverIO [66], as a general model for multimodal data, has demonstrated its capabilities in remote sensing image segmentation when combined with 3D convolutions [74].

3. Dataset and Methods

3.1. Dataset Description

The Tinto dataset [9] is a point cloud dataset with additional hyperspectral information. The respective scene is located in the open-pit mine Corta Atalaya, Minas de Riotinto, Spain. There are three hyperspectral feature groups, i.e., Long-Wave Infrared (LWIR) with 126 bands, Short-Wave Infrared (SWIR) with 141 bands, and Visible and Near Infrared (VNIR) with 51 bands.

As mentioned before, we combined hyperspectral data with 3D point clouds, which is rare for common point cloud datasets. We utilized drone-borne and terrestrial sensors for the data acquisition. An aerial camera on the drone captured the RGB data. From there, dense 3D RGB point clouds were reconstructed using Structure from Motion (SfM) and a multi-view stereo technique. Meanwhile, hyperspectral data were captured using terrestrial sensors. Subsequently, the data were back-projected onto the point clouds to incorporate the additional hyperspectral features.

The dataset consists of 10 mineral classes: Saprolite, Chert, Sulfide, Shale, Purple Shale, Mafic A, Mafic B, Felsic A, Felsic B, and Felsic C. It has 297,968 points in the training set and 2,885,178 points in the testing set. The training samples were collected manually from the field by domain experts, ensuring the highest standards of accuracy. These samples were then meticulously evaluated in the laboratory to determine the precise classification

of the minerals. Due to the challenging conditions at the mining site, 10% of the overall data available was collected. We can see the distribution of training and testing sets in Figure 1 where the training set is colored red and the testing set is colored blue. We can also see the class distribution of the training set. As we may notice, there is an imbalance class issue, which makes this dataset more challenging.

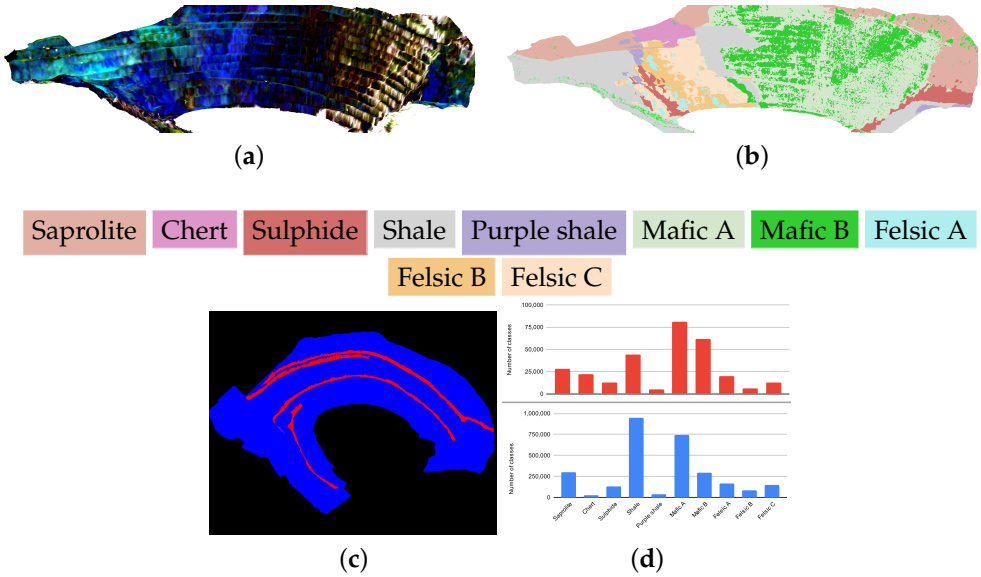

Figure 1. (**a**) False-color visualization of the LWIR hypercloud data at 10,114, 9181, and 8545 nm, (**b**) the corresponding semantic label, (**c**) training (red) and testing (blue) sets, and (**d**) the class distribution of training (red) and testing (blue) sets.

3.2. Data Preprocessing

Assuming both training and testing sets of point clouds denoted by P, we initially split these point clouds into blocks referred to as P_{block}. The size of each block is fixed. It is crucial to choose an appropriate block size that is large enough to capture the context of the points within the block while avoiding memory overloading.

To ensure that each block contains sufficient information, we set a threshold value as the minimum number of points required for each block. Before the training phase, we remove all blocks that contain a number of points below the threshold value. This step is essential to ensure that the network is trained on sufficient number of points.

After filtering the point clouds, we sample the points in each remaining P_{block} to obtain a fixed number of points, which is denoted by n. We implement the following conditional operation to check whether the number of points in each P_{block}, denoted as s, is greater or less than n. If s is less than or equal to n, we duplicate the points to obtain n points, while if s is greater than n, we downsample the points to obtain n points.

$$P_{block} = \begin{cases} \text{duplicate} & \text{if } s \leq n \\ \text{downsample} & \text{if } s > n \end{cases} \tag{1}$$

The choice of n depends on the point density of the point clouds itself. If it is set too low, then we lose the resolution of the data. But if it is set too high, then many points are duplicated, leaving the network to learn synthetic and meaningless features.

In summary, this preprocessing method ensures that the neural network trains on the same size of the input tensors for multi-batch multi-GPU training while reducing the computational complexity of the network.

The purpose of blocking and sampling points here is to have a consistent input size for the network during the training. It is similar to image-based networks where the input is always having a fixed size, e.g., 256×256, 512×512 pixels, or so on. Since the training phase is on batch processing, all batches need to have the same size if we use multiple

batch sizes. During the testing, it is not mandatory to sample the number of points if we only use a batch size equal to one. Instead, we have to avoid sampling because we want to classify all points. However, it still needs to be split into blocks with the same size as in the training to ensure that the contextual information of the blocks is not different.

We observed that the 50×50 meters block size covers sufficient semantic information while keeping the block size reasonable for the computation cost. This size was also chosen to maximize the number of points per block. However, it should be noted that different datasets might need different sizes, depending on the object sizes and scene complexities. We then created blocks in 1000 random locations on the training set instead of generating non-overlapping systematic blocks as performed in previous works [47,49]. Next, the number of points for each block was set to 4096 points to have a similar order of magnitude between different blocks with different point density. Therefore, the downsample and upsample will only have a minor effect.

3.3. Proposed Network

Our network architecture utilizes several techniques to achieve its performance. The complete architecture can be seen in Figure 2. In this section, we will provide a brief overview of these techniques.

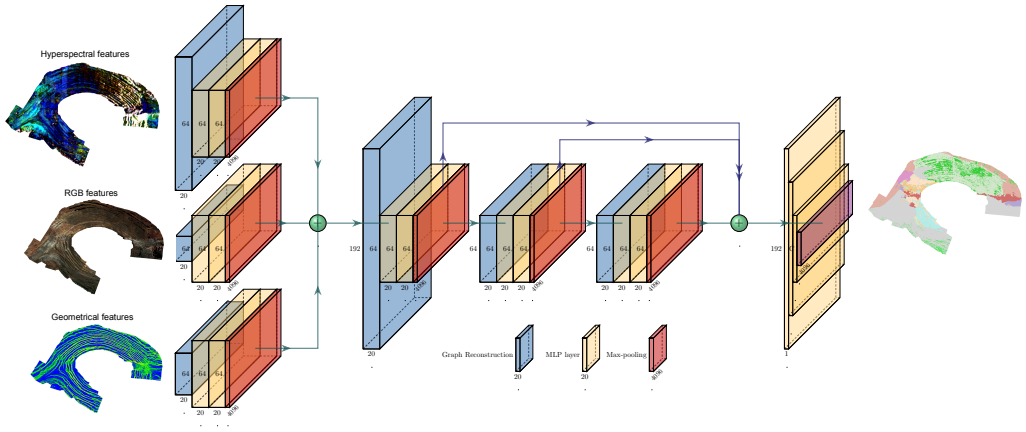

Figure 2. Illustration of the multi-stream network. The architecture consists of 3 individual streams in the earlier part of the network, which is followed by a typical single-stream network with skip connection links (illustrated as blue arrows) to bring the information from the earlier layers to the latter. The 3 individual streams are designed to work with 3 different modalities. Those are hyperspectral, RGB, and geometrical data. The channel-wise concatenation is utilized for merging the output of the 3 different streams. It is also used for concatenating the skip connection links. Graph reconstruction, MLP, and max-pooling layers are stacked together as the features encoder block. This block is utilized in each stream separately and then repeated three times for deeper layers. Finally, dense layers are applied in the last layers for the classification. Please note that height represents the number of features, width represents the number of k-NN points, and depth represents the number of points for each block.

3.3.1. Graph Reconstruction

In our previous work [9], we extensively analyzed various encoder architectures and concluded that EdgeConv (DGCNN) consistently delivers strong performance for mineral classification with hyperspectral data. Notably, EdgeConv's effectiveness is complemented by its lightweight operation, which is attributable to its reliance on graph MLP for feature aggregation. By employing EdgeConv, we ensure the equal treatment of different modalities, enabling the network to learn the amalgamation of features from all modalities. Further elaboration on this aspect is provided in the subsequent section.

EdgeConv layer is a graph-based operation. Suppose a point cloud P with d-dimensional features and p points. A point cloud P is then represented as $\{P_i \mid i = 1, \ldots, p\} \in \mathbb{R}^{p \times d}$.

The simplest point cloud always has $d = 3$ from the XYZ coordinates. The hyperspectral point clouds, however, have d equal to the number of hyperspectral channels plus the 3-dimensional of the coordinates itself. Therefore, P_i of each hyperspectral point is a vector $\mathbf{v} = (v_1, \ldots, v_{d-1}, v_d)$ where $v_1 = X\,coordinate, v_2 = Y\,coordinate, v_3 = Z\,coordinate$.

The EdgeConv layer creates a directed graph $G = (V, E)$ with V as vertices and E as edges. Since the graph allows edges to loop, the E is defined as $E \subseteq \{(x, y) \mid (x, y) \in V^2\}$ where (x, y) is the edge directed from x to y.

For each block, the graph G is constructed by finding the k-nearest neighbor (k-NN) of the point cloud P_{block} with the number of points n. It should be noted that n is the chosen number for the point size of the blocks. The distance to the k-NN could be defined as a Euclidean distance in the metric space (X, Y, Z) on simple point clouds plus the Euclidean distance in the feature space for the higher dimensionality data as in hyperspectral point clouds. In our implementation, the distance is calculated from both of them. However, only the normalized XYZ is used in the metric space to ensure that the distances are in the same range for all features.

Suppose a set of neighboring points where P_i is the center point and P_{ij} represents the k nearest points where $\{P_{ij} \mid j = 1, \ldots, k\} \in \mathbb{R}^{k \times d}$; then, the edges are simply defined as (P_i, P_{ij}). With Θ as a set of trainable weights and $h_\Theta : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ as a non-linear function, EdgeConv [14] defines the edge features as $e_{ij} = h_\Theta(P_i, P_{ij})$. Explicitly, if $\Theta = (\theta_1, \ldots, \theta_m, \phi_1, \ldots, \phi_m)$, the output of the EdgeConv is expressed in Equation (2).

$$e'_{ijm} = (\theta_m \cdot (P_{ij} - P_i) + \phi_m \cdot P_i) \tag{2}$$

In a point cloud segmentation task, it is important to capture both local and global structures to learn extensive point features. To this end, EdgeConv extracts local neighborhood structures by $P_{ij} - P_i$ and combines them with the global features captured by P_i.

Suppose a tensor is taken as an input with the size of (b, d, n) where b is the batch size, d is the number of features, and n is the number of points in a block; then, the output of the graph features will be a tensor with the size of $(b, d \times 2, n, k)$ where k is the number of point neighbors.

3.3.2. Multi-Stream Architecture

Classification tasks with different modalities of the input data have frequently been investigated in image segmentation tasks by implementing multi-stream networks as in [75–77]. The idea of the multi-stream network is to let the network learn different features for different data modalities.

In contrast to learning multimodal data through multiple sub-networks, simply concatenating all different modalities to the same features vector is the naive solution. It treats all modalities equally, although some modalities are probably more informative than others. It leads to a dilution of useful information, as the simple concatenation may include irrelevant features [78,79].

In that sense, we argue that it would be beneficial to have a multi-stream network since our dataset has at least three different data types: XYZ, hyperspectral, and RGB. We present our proposed network in Figure 2. In this architecture, instead of stacking all features together and having a single-stream network, as in data-level fusion, the network is expanded into three independent streams. Despite its advantage, the construction of a large and heavy network is avoided, considering that the single stream network of hyperspectral features is already acknowledged as heavy. Dividing the network into three streams would ultimately triple the network's size. In that sense, the network is divided into three streams only for the first part. The output of each stream is then concatenated as the input for the second part of the network. While it would have been possible to have three fully independent streams and merge the features in the last layer, this approach would have led to a significant increase in the number of parameters. In our approach, the learning of different features from different data modalities is facilitated by the network without resulting in an excessive number of parameters.

In our model, an EdgeConv block is utilized for each stream. This block comprises a graph layer, two consecutive MLP layers, and max-pooling layers. In Figure 2, each layer is represented by a different color. The graph layer computes edge features, as explained in the previous section. The MLP layers act as feature encoders, each with 64 filters. These MLP layers are then followed by a max-pooling layer that operates on k-NN points. This means that the max-pooling layer selects the most significant points in the neighborhood. It is important to note that we define the distance for k-NN not only by the physical distance but also by the feature distance. The distance is computed using the Euclidean distance in multidimensional data. To enhance efficiency, the calculation is optimized in PyTorch version 1.13.0.

Each stream in the model produces a tensor of size (b, l, n), where b is the batch size, l is the number of filters of the MLP layer, and n is the number of points. To combine these streams, channel-wise concatenation is employed to create a tensor of size $(b, l \times 3, n)$, as shown in Figure 2. Here, the height represents the number of features, and the second graph layer in the network has a height three times the height of each previous max-pooling layer. After the concatenation, a single stream network is applied to process all the features. This network consists of three identical EdgeConv blocks stacked consecutively, each with a graph layer, two MLPs, and a max-pooling layer. Additionally, shortcut connections are used to obtain multi-scale features, again using channel-wise concatenation. The architecture does not use a downsampling mechanism, which is similar to PointNet [47]. As a result, the n number of points, or the block size in our case, is maintained for every layer, avoiding the downsampling and upsampling mechanisms seen in other semantic segmentation architectures [13,49], as illustrated in Figure 2.

Our architecture differs from the original Dynamic Graph CNN [14], as a global pooling layer is not used. The global pooling layer is commonly used to achieve permutation invariance given a set of points in object or shape classification tasks [47]. However, we argue that this pooling layer is not suitable for semantic segmentation tasks where points may have different labels.

The reason for this is that the pooling layer attaches the same pooled features to all points, which is useful for classification but not for segmentation. In segmentation tasks, the original encoded features of each point need to be maintained to preserve its semantic information. Therefore, the global pooling layer is avoided in our architecture for semantic segmentation tasks.

3.3.3. The 3D CNN as Channel-Attention Module

We enhance our model's ability to leverage spectral information by integrating a channel attention module. This module is designed to dynamically emphasize relevant features within the hyperspectral data. Our implementation strategy involves embedding 3D convolutional neural networks (CNNs) on top of the EdgeConv operator, which is a key component within the hyperspectral stream of the network. This innovative approach draws inspiration from the Efficient Channel Attention Network (ECANet) [19] that employs 1D CNNs as gating mechanisms to selectively modulate channel-wise features, effectively enhancing the model's discriminative power. By adapting this concept to our hyperspectral classification task, we aim to improve the model's ability to focus on salient spectral features, thereby boosting the overall classification performance.

Formally, the output matrix $\mathbf{O} \in \mathbb{R}^{b \times c_{out} \times d_{out} \times h_{out} \times w_{out}}$ of a 3D convolutional layer with input $\mathbf{I} \in \mathbb{R}^{b \times c_{in} \times d_{in} \times h_{in} \times w_{in}}$ can be described as

$$\mathbf{O}_{\mathbf{b_i, c_{out}}} = \mathbf{B}_{\mathbf{c_{out_j}}} + \sum_{\bigtriangledown=0}^{c_{in}-1} \mathbf{W}_{c_{out_j}, \bigtriangledown} \bigstar \mathbf{I}_{b_i, \bigtriangledown} \tag{3}$$

where $\mathbf{W}$ is a weight matrix, $\mathbf{B}$ is a bias, and $\bigstar$ is a 3D cross-correlation operator with element-wise matrix multiplication.

While a 3D convolutional filter can extract spatial–spectral features, it is only suitable for grid-like data such as hyperspectral data in raster format and cannot be directly applied

to hyperspectral point clouds. On the other hand, the EdgeConv layer works well for capturing local spatial structures, but it ignores the spectral patterns of the point clouds. To incorporate both spatial and spectral information, we need to integrate the 3D convolutional filter with the EdgeConv layer in the architecture. Here, we explain how we combine the 3D convolutional filter with the EdgeConv layer to achieve this integration.

First, graph reconstruction is performed on k-NN on points to gain the benefit of the local structures. Given n points and k nearest-neighbors for each point, the graph is a tensor of size $(b, d \times 2, n, k)$ with b as the batch size and d as the dimension of the features. We then add an empty dimension, so the tensor has a size of $(b, 1, d \times 2, n, k)$. Then, two identical 3D convolutional blocks are used; each has 4 filters with size $(32, 1, 1)$. As it suggests, the filters learn the spectral features by capturing the pattern for every 32 hyperspectral features. The latter $(1, 1)$ dimension works similarly to fully connected layers, as in MLP. The output tensors will be in the size of $(b, 4, d \times 2, n, k)$. In this manner, we can exploit the benefit of hyperspectral features on point cloud data. Figure 3 illustrates the operation of the hyperspectral stream using 3D CNNs.

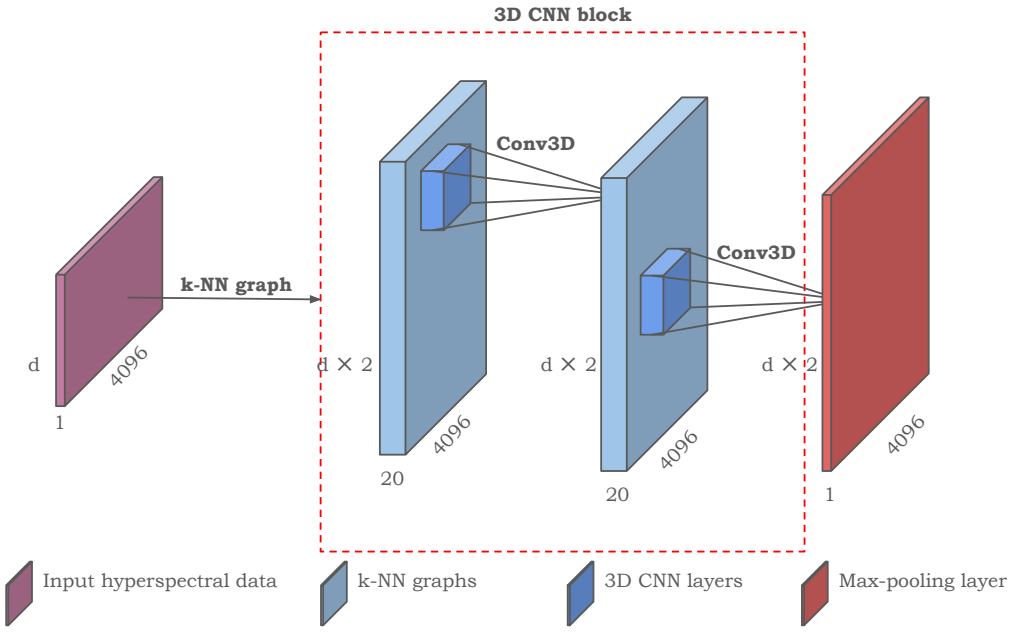

Figure 3. Schematic diagram of the 3D CNN block in the hyperspectral stream.

Subsequently, the dimension is reduced into $(b, d \times 2, n)$ by applying two max-pooling operators. The former is used to aggregate significant features among all filters, and the latter is used to aggregate the significant points in the neighborhood. Finally, the MLP-like network is used, as in other streams.

3.3.4. Geometric Features Transformation

In our experiments, we found that the graph network directly learning geometric features from the XYZ coordinates is not beneficial for detecting the mineral classes in the open-pit mine area. This is because minerals do not have distinct shapes that differ from one to another. Moreover, the training area spreads only in the horizontal direction, while the testing area has flat and sloped terrain, which is similar to a typical landscape in an open-pit mine. We provide more details about the dataset in Section 4.

However, we avoid completely ignoring the XYZ coordinates and still expect to gain some benefit from them. To further exploit the XYZ coordinates, we follow the Superpoint Graph (SPG) network [80] by transforming the XYZ coordinates into pre-computed geometrical features. These features capture the geometric characteristics of each point with respect to its neighbors, such as whether the point lies on a planar surface or

not. Then, the geometric features are fed to the network as one of the streams. Unlike SPG where the geometric features are computed on the superpoint, the geometrical features for each point are computed before the graph reconstruction in our work. We followed the 14 geometric features as suggested by [81].

Let $\lambda_1 > \lambda_2 > \lambda_3 > 0$ be the eigenvalues and $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ be the eigenvectors of the covariance matrix of points to the neighbors; the 14 geometric features are formally described by [81] as in Table 1.

Since the computed features are highly dependent on the definition of the neighborhood, it is crucial to choose it carefully. In our architecture, a ball query is preferred to define the neighborhood rather than a *k*-NN. The reason behind this choice is that the ball query maintains the spatial extent of the neighborhood, while the spatial extent of the *k*-NN varies depending on the local density of the point cloud. Additionally, two different radii, 0.5 m and 1 m, are selected to capture multi-scale features. Overall, the point clouds are transformed into 28 geometrical features through the combination of EdgeConv blocks and multiscale features.

Table 1. The 14 transformed geometric features.

No	Feature	Description
1	Sum	$\lambda_1 + \lambda_2 + \lambda_3$
2	Omnivariance	$(\lambda_1 \cdot \lambda_2 \cdot \lambda_3)^{\frac{1}{3}}$
3	Eigenentropy	$-\sum_{i=1}^{3} \lambda_i \cdot ln(\lambda_i)$
4	Anisotropy	$(\lambda_1 - \lambda_3)/\lambda_1$
5	Planarity	$(\lambda_2 - \lambda_3)/\lambda_1$
6	PCA 1	$\lambda_1 - (\lambda_1 + \lambda_2 + \lambda_3)$
7	PCA 2	$\lambda_2 - (\lambda_1 + \lambda_2 + \lambda_3)$
8	Linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
9	Surface Variation	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$
10	Sphericity	λ_3/λ_1
11	Verticality	$1 - \lvert\langle[001], \mathbf{e}_3\rangle\rvert$
12, 13, 14	3 normal vectors	N_x, N_y, N_z

3.4. Comparison to Image Segmentation

The hyperspectral attributes were not originally part of the point clouds but were derived from hyperspectral data obtained by different sensors. These additional attributes were added by backprojecting the hyperspectral data onto the point clouds. It would be interesting to compare the performance of image-based and point-based deep learning networks by examining the segmentation results of both datasets.

To ensure a fair comparison between point cloud and image-based segmentation, we developed an image segmentation network based on the graph network architecture used for point cloud segmentation. The approach is similar with edge features computed for each pixel to its neighboring pixels using a fixed size kernel instead of *k*-NN to define the neighborhood and construct the graph. Specifically, the edge features are calculated as $e_{ij} = h_\theta(x_i, x_j - x_i)$, where x_i represents the central pixel and x_j represents the surrounding pixel. To obtain multi-scale graph features with different receptive field sizes, two different kernel sizes are used, 3×3 and 5×5 pixels. Once the graph is reconstructed, it passes through two shared MLP layers for each scale, and the results are concatenated before feeding them into the fully connected layers. Figure 4 illustrates the graph reconstruction and the architecture of the 2D-based network.

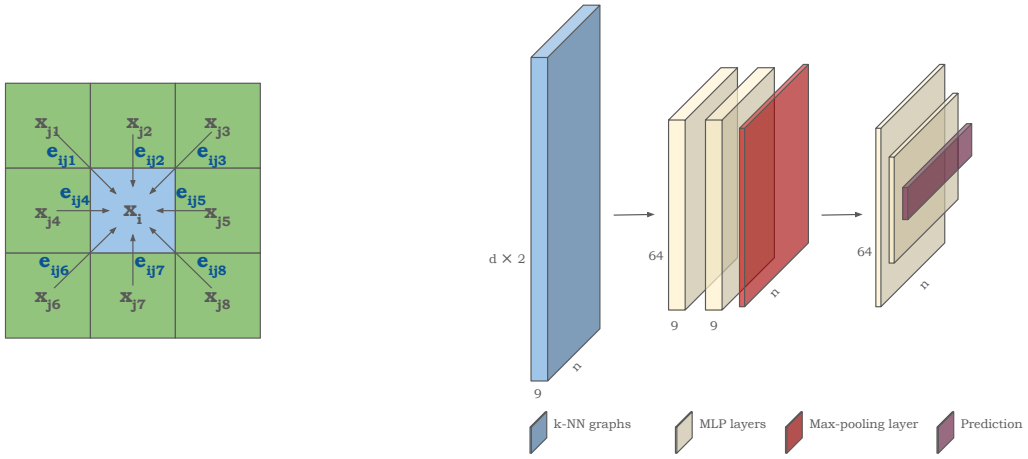

Figure 4. (**Left**): Graph reconstruction using an image (x_i represents the central pixel, x_j represents the surrounding pixel, and e_{ij} denotes the edge features). (**Right**): Illustration of the 2D-based network.

In addition to the graph-based network, we also train a simple MLP network that does not consider the relationship between pixels and their neighborhoods. This allows the network to learn solely from the spectral values of individual pixels.

4. Results

4.1. Training Setting

The network was trained from scratch using the SGD optimizer with momentum [82] and implementing it on Pytorch. The momentum was set to 0.9 with a learning rate of 0.001. A training batch size of 56 was used, and the network was trained for 100 epochs. The best models were selected based on the validation set.

A GPU cluster with 8 Nvidia A100s was utilized to speed up the training time; however, it is not necessary to use such a huge machine to train our network, as the size of the network and the dataset are reasonable to be trained with the regular GPUs. Depending on the input of the hyperspectral data, the numbers of parameters of the network are 311,424, 324,112, and 305,424 for the LWIR, SWIR, and VNIR sensors, respectively.

The class weights were also added during training to address the class imbalance. Given the number of instances for each class n_c and the number of all instances n_t, we computed the weight for each class w_c according to Equation (4) and passed the weights to the cross-entropy loss function.

$$w_c = 1 - \left(\frac{n_c}{n_t}\right) \tag{4}$$

The code for our work will be made available at https://github.com/aldinorizaldy/HyperspectralGraphNetwork, accessed on 7 May 2024.

4.2. Quantitative Results

The performance of our method is compared to the baseline methods as reported in our prior work [9]. Regarding the evaluation metric, in addition to the overall accuracy (OA), the F1-score per-class accuracy is used to show the detailed accuracy for each class. Furthermore, the class-averaged mean-F1 score (mF1) is also reported, with balanced (bal-mF1) and unbalanced (unbal-mF1) scores, to have a more comprehensive comparison of the performance of our network against the baselines.

In terms of the OA, as seen in Tables 2–4, the proposed method outperformed all comparative methods by a large margin, i.e., 19.2, 4.4, and 5.6 percentage points for each LWIR, SWIR, and VNIR dataset, respectively. In terms of the mean-F1 score, our method also shows similar results by surpassing the existing methods by 16.8, 4.0, and 4.8 percentage points for balanced scores and 16.6, 5.1, and 5.8 percentage points for unbalanced scores.

Table 2. Accuracy of LWIR dataset. The highest accuracy is indicated in bold.

Class	PointNet	PointNet++	PointCNN	ConvPoint	DGCNN	PT	PCT	Ours
Saprolite	38.5	39.1	46.0	41.9	39.4	40.5	39.3	**68.1**
Chert	16.3	17.4	17.5	18.4	17.8	25.9	17.8	**43.0**
Sulfide	50.1	50.0	45.9	46.8	45.9	41.8	49.3	**68.2**
Shale	45.5	42.4	58.7	47.9	46.8	44.5	48.1	**79.9**
Purple Shale	7.2	11.3	9.4	10.8	7.8	6.0	8.6	**27.0**
Mafic A	56.2	60.4	55.5	57.8	58.9	63.1	57.5	**68.9**
Mafic B	**40.8**	38.7	40.7	39.0	37.6	33.9	36.4	39.1
Felsic A	24.2	20.2	27.4	31.7	18.9	24.0	18.9	**52.7**
Felsic B	**20.6**	19.5	14.8	15.5	17.5	26.7	19.0	18.7
Felsic C	35.8	33.5	23.9	38.0	28.3	24.0	30.2	**48.2**
OA	43.2	43.0	47.3	45.1	42.7	45.3	43.3	**66.5**
bal-mF1	43.9	43.5	48.2	45.7	43.9	44.2	44.1	**65.0**
unbal-mF1	33.5	33.2	34.0	34.8	31.9	33.0	32.5	**51.4**

Table 3. Accuracy of SWIR dataset. The highest accuracy is indicated in bold.

Class	PointNet	PointNet++	PointCNN	ConvPoint	DGCNN	PT	PCT	Ours
Saprolite	55.0	63.9	49.6	58.9	53.7	51.0	66.4	**72.3**
Chert	32.7	**44.2**	15.3	21.0	37.2	26.8	43.7	43.9
Sulfide	51.5	54.4	36.6	54.3	47.9	29.0	49.6	**63.2**
Shale	72.3	76.4	73.7	74.3	76.3	71.8	78.1	**82.0**
Purple Shale	27.8	14.8	10.8	11.7	16.3	6.9	20.2	**36.8**
Mafic A	79.2	79.2	75.3	79.5	78.5	74.5	77.3	**81.6**
Mafic B	**77.8**	76.4	49.8	76.3	76.6	42.6	74.7	73.8
Felsic A	62.3	66.6	67.1	59.4	61.2	54.9	66.8	**72.3**
Felsic B	47.8	**59.0**	4.3	42.9	52.9	2.7	52.7	58.9
Felsic C	71.9	72.0	44.0	64.8	69.3	43.2	66.4	**72.9**
OA	67.7	71.5	59.7	67.7	68.9	60.2	71.8	**76.2**
bal-mF1	69.5	72.3	62.0	69.6	70.1	59.6	71.9	**76.2**
unbal-mF1	57.8	60.7	42.7	54.3	57.0	40.3	59.6	**65.8**

Table 4. Accuracy of VNIR dataset. The highest accuracy is indicated in bold.

Class	PointNet	PointNet++	PointCNN	ConvPoint	DGCNN	PT	PCT	Ours
Saprolite	60.3	64.0	52.7	55.1	62.8	43.2	68.5	**82.4**
Chert	23.1	29.3	14.3	13.6	28.9	25.4	28.7	**35.3**
Sulfide	51.9	47.3	39.5	49.0	51.0	25.4	49.4	**55.5**
Shale	69.4	62.0	62.1	57.3	62.6	59.5	65.5	**73.2**
Purple Shale	38.0	**43.4**	23.7	19.6	30.5	4.8	34.4	40.9
Mafic A	73.1	73.8	63.0	69.6	70.2	70.2	72.1	**76.6**
Mafic B	**54.4**	51.1	42.2	46.9	48.7	18.8	46.8	53.9
Felsic A	57.7	53.9	**57.9**	53.4	52.0	42.5	57.2	56.2
Felsic B	33.2	37.1	19.1	37.7	34.0	2.3	33.4	**40.9**
Felsic C	51.2	52.0	28.1	51.3	50.6	34.5	50.5	**56.8**
OA	62.1	60.7	51.8	55.4	59.1	51.2	61.2	**67.7**
bal-mF1	63.5	61.2	54.0	56.7	59.8	49.8	61.9	**68.3**
unbal-mF1	51.2	51.4	40.3	45.4	49.1	32.7	50.6	**57.2**

In the LWIR dataset, our method achieved an OA of 66.5%, while PointNet, PointNet++, PointCNN, ConvPoint, DGCNN, PointTransformer, and PCT achieved OAs of 43.2%, 43.0%, 47.3%, 45.1%, 42.7%, 45.3%, and 43.3%.

Similarly, in the SWIR dataset, our proposed method achieved an OA of 76.2%, while PointNet, PointNet++, PointCNN, ConvPoint, DGCNN, PointTransformer, and PCT achieved OAs of 67.7%, 71.5%, 59.7%, 67.7%, 68.9%, 60.2%, and 71.8%, respectively. Furthermore, our proposed network also shows its superiority on the third dataset, the VNIR dataset, where it obtained an OA of 67.7% while PointNet, PointNet++, PointCNN, ConvPoint, DGCNN, PointTransformer, and PCT achieved OAs of 62.1%, 60.7%, 51.8%, 55.4%, 59.1%, 51.2%, and 61.2%.

Our method also consistently achieved the highest class-averaged mean-F1 scores both for the balanced and unbalanced scores. The balanced mean-F1 shows the weighted accuracy of all classes with respect to the number of instances for each class. Meanwhile, the unbalanced mean-F1 simply averaged all per-class accuracy without taking into account the class imbalance. The proposed network obtained 65.0%, 76.2%, and 68.3% of the balanced mean-F1 and 51.4%, 65.8%, and 57.2% of the unbalanced mean-F1 for the LWIR, SWIR, and VNIR datasets, respectively. It is expected to have lower values for the unbalanced scores since some classes have a limited number of samples.

These results prove the ability of our method to infer the semantic information of mineral classes, which is measured by not only the overall point accuracy but also more importantly the class-averaged accuracy. Furthermore, if we look at the two classes with the smallest number of samples, Chert and Purple Shale, the proposed method achieved the highest per-class F1-score accuracy on all datasets. It only loses by PointNet++ on the VNIR dataset for the Purple Shale class.

4.3. Qualitative Results

We further compare the results of different methods by visually inspecting the prediction. Figures 5–7 plot the point clouds, color-coded by the predicted labels, on the LWIR, SWIR, and VNIR datasets.

As seen in Figure 5, our method generally resulted in less noisy labels than the comparison methods. Furthermore, all the comparison methods suffered from false positive errors of the Saprolite and Chert class on the left side of the area. Our method, on the other hand, predicted this particular area with the correct Shale class. Nevertheless, none of them has successfully predicted the Purple Shale class on the right side of the area.

In the SWIR dataset, as illustrated in Figure 6, our method also demonstrated superiority over the existing methods. We can clearly observe that our method avoids the false positive error of the Saprolite class, as can be seen in the results of the comparison methods. On the other hand, the existing methods suffered from the false negative error of the Sulfide class, while our method predicted this particular class more accurately.

The results in the VNIR dataset, however, show different errors where almost all the methods failed to predict the Shale class and wrongly labeled the area with the Felsic A class. Point Transformer, on the other hand, has the minimum false positive error of the Felsic A class, but it almost completely failed to detect the Mafic B class.

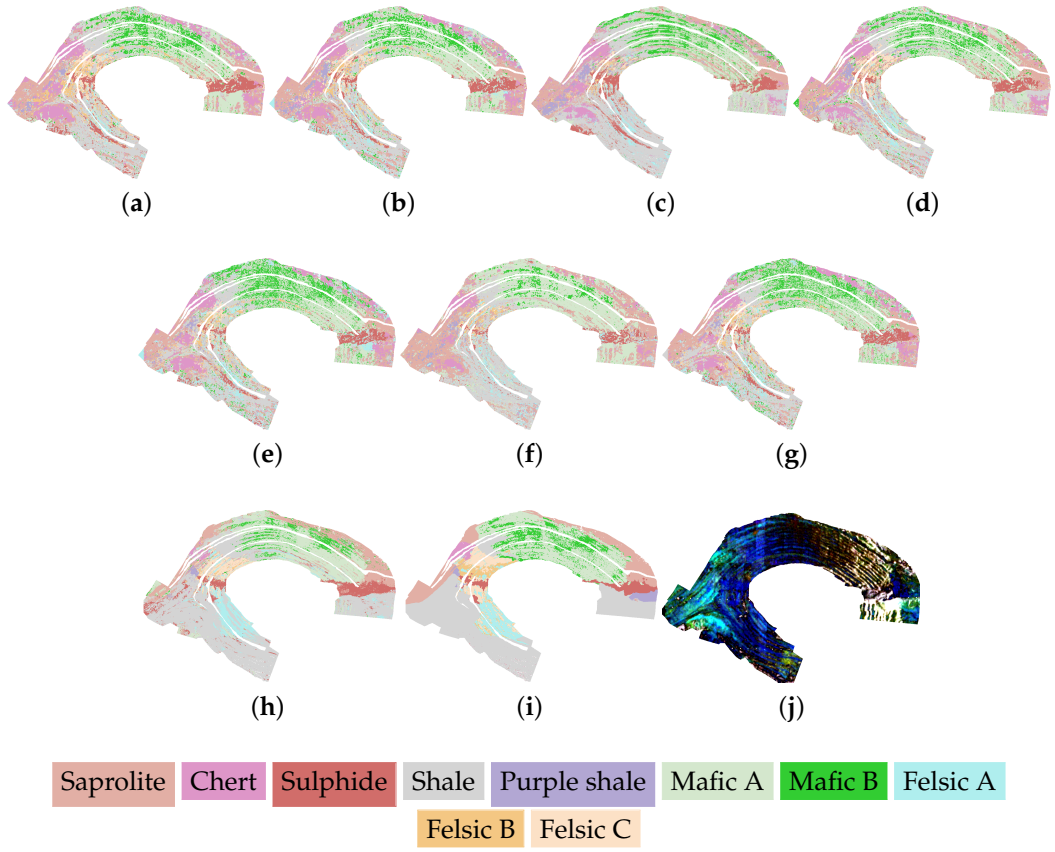

Figure 5. The predicted point clouds of the LWIR dataset and the corresponding ground truth. (**a**) PointNet; (**b**) PointNet++; (**c**) PointCNN; (**d**) ConvPoint; (**e**) DGCNN; (**f**) Point Transformer; (**g**) PCT; (**h**) ours; (**i**) ground truth; (**j**) LWIR hypercloud.

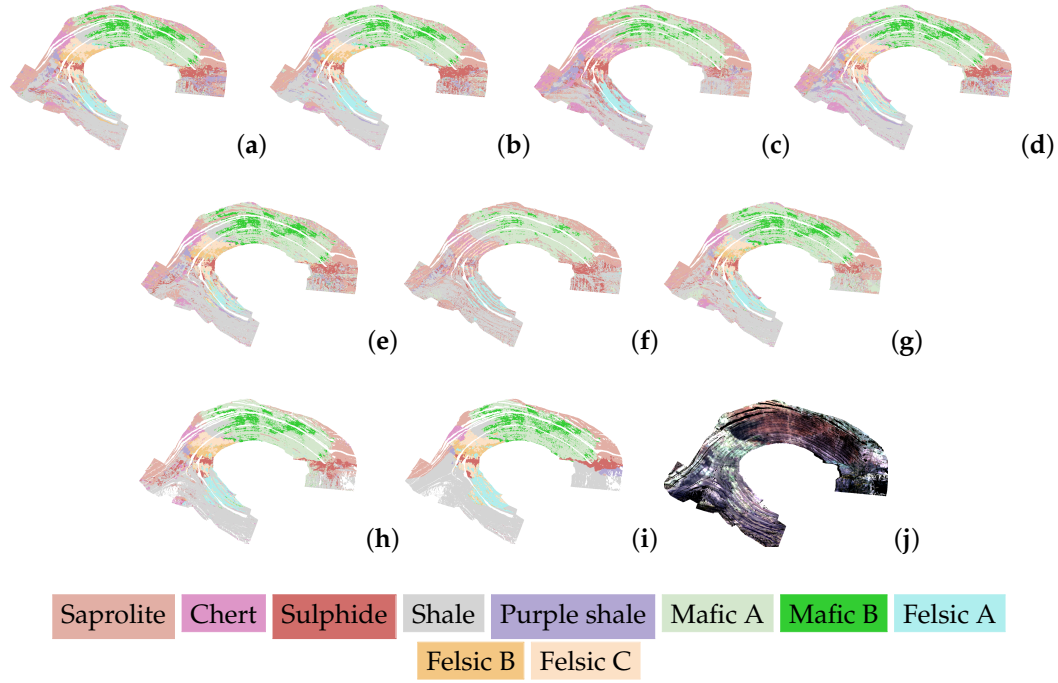

Figure 6. The predicted point clouds of SWIR dataset and the corresponding ground truth. (**a**) PointNet; (**b**) PointNet++; (**c**) PointCNN; (**d**) ConvPoint; (**e**) DGCNN; (**f**) Point Transformer; (**g**) PCT; (**h**) ours; (**i**) ground truth; (**j**) SWIR hypercloud.

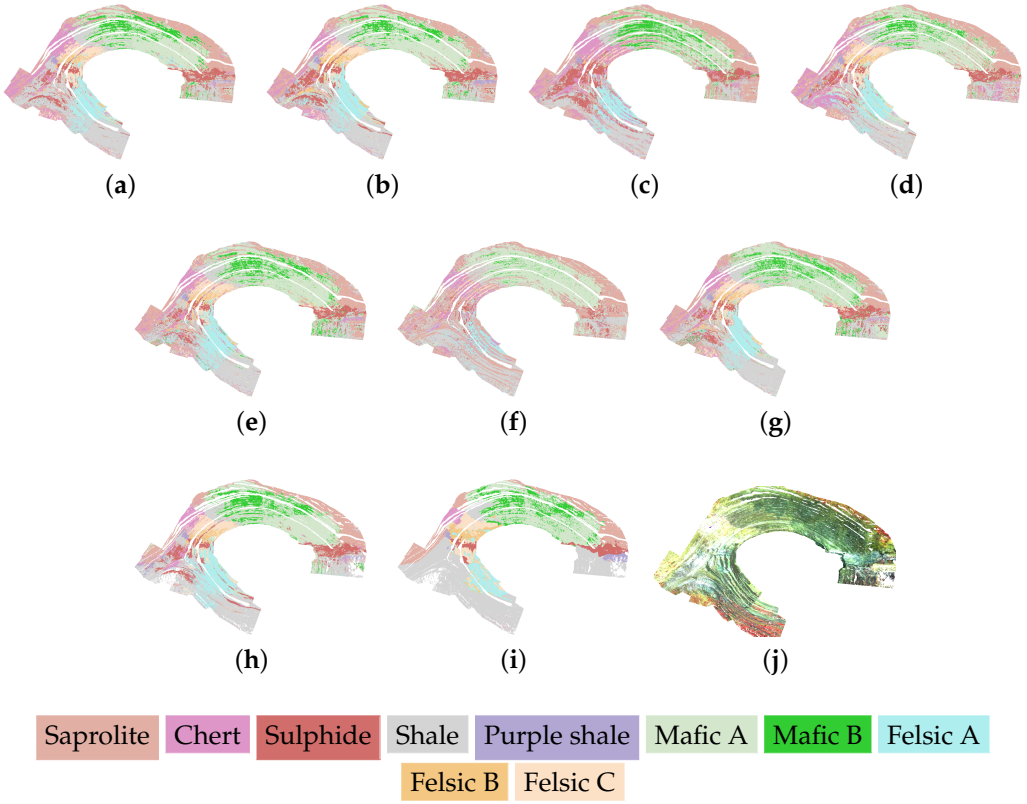

Figure 7. The predicted point clouds of VNIR dataset and the corresponding ground truth. (**a**) PointNet; (**b**) PointNet++; (**c**) PointCNN; (**d**) ConvPoint; (**e**) DGCNN; (**f**) Point Transformer; (**g**) PCT; (**h**) ours; (**i**) ground truth; (**j**) VNIR hypercloud.

4.4. Comparison of Point Cloud and Image Segmentation

After obtaining the hyperspectral attributes of the point clouds from the corresponding images, it is intriguing to observe the image-based segmentation's performance on the same dataset. We acknowledge that point clouds are in 3D format, providing a superior perception of the data to the network. However, deep learning has made significant strides in the field of image-based segmentation. Therefore, comparing the performance of the two methods can shed light on the ideal approach for achieving greater accuracy when both datasets are available. In Table 5, we show the F1-score per-class accuracy, overall accuracy, and F1-score of the image-based segmentation and point-based segmentation using the SWIR dataset.

Table 5. Comparison of image-based to point-based segmentation using the SWIR dataset.

Class	Image-Based Graph	Image-Based MLP	Point-Based Graph
Saprolite	31.6	–	72.3
Chert	72.8	56.3	43.9
Sulfide	42.1	–	63.2
Shale	67.1	31.2	82.0
Purple Shale	56.6	–	36.8
Mafic A	60.9	–	81.6
Mafic B	60.4	–	73.8
Felsic A	–	–	72.3
Felsic B	81.9	32.8	58.9
Felsic C	77.8	21.5	72.9
OA	67.3	37.1	76.2
bal-mF1	63.6	26.8	76.2
unbal-mF1	55.1	14.1	65.8

Table 5 highlights that not all classes are accurately predicted in the image segmentation result of the MLP network, whereas graph-based image segmentation and point-based segmentation predict all classes successfully. Although some classes of the image segmentation results display higher per-class accuracy, the whole point accuracy OA and the class-averaged accuracy F-1 scores of image segmentation are lower than those of point cloud segmentation.

Figure 8 shows the predicted images and the corresponding ground truth of the graph-based and MLP networks. As shown, the graph-based network clearly outperformed the MLP network.

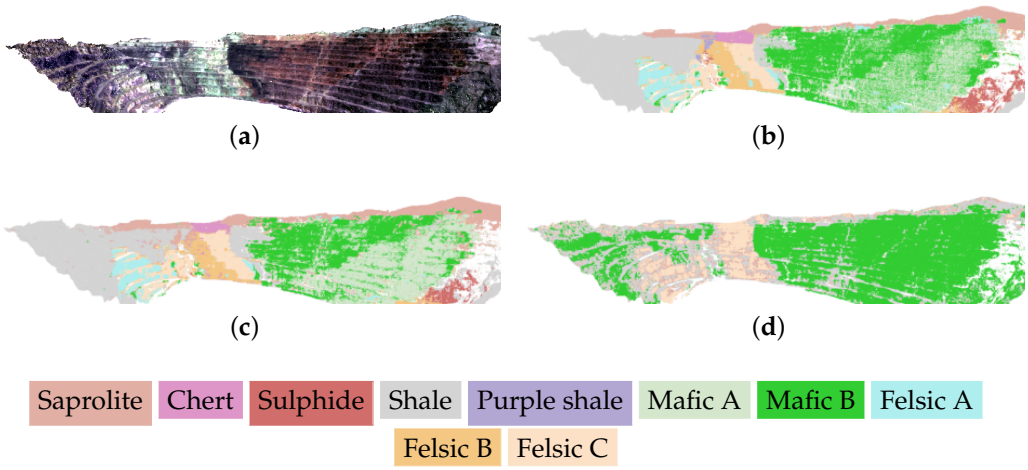

Figure 8. Image-based segmentation results using the SWIR dataset. (**a**) SWIR; (**b**) ground truth; (**c**) graph-network; (**d**) MLP.

5. Discussion

Our method relies on utilizing multimodal data on a multi-stream network. Here, we discuss the significance of each component of our method. We began by exploring the importance of integrating multiple modalities, which was followed by evaluating the significance of the multi-stream network design. Furthermore, we also reported our findings regarding the implementation of 3D CNNs as channel attention modules, demonstrating that integrating 3D CNNs significantly enhances the classification results.

5.1. Combining Different Data Modalities

We investigated the significance of combining various data modalities to the accuracy of the prediction. The DGCNN-like network was chosen as the basis of our experiments. We started the experiments with only hyperspectral features and added more features to examine which features vector results in the highest accuracy. The comparison results are shown in Table 6.

Our experiments are shown in Table 6 and suggest that combining more features resulted in higher segmentation accuracy. The benefit is more significant on the LWIR dataset, where the hyperspectral or RGB features alone are not sufficient to exceed 50% of the overall accuracy. Combining both features improves the result by 8.54 percentage points. Furthermore, adding the extra geometrical features boosts the OA by 3.40 percentage points. The results are consistent when fusing hyperspectral and RGB features in the SWIR dataset. The model's accuracy improves by a small margin (0.08 percentage points). Adding geometric features, however, does not help to increase the accuracy. Instead, it slightly reduces the accuracy. However, we argue that combining all data through feature-level fusion, such as using techniques like multi-stream networks, yields higher accuracy compared to combining data through data-level fusion as demonstrated in this section. In the subsequent section, we present evidence where all available data are leveraged, and fusion occurs after feature extraction in separate sub-networks. This configuration outperforms various configurations presented earlier.

Figure 9 shows the benefit of adding more modalities to the networks. The greater the number of modalities of data included in the networks, the less noise that is visible in the prediction.

Table 6. Different configuration of features as input to the network. The highest accuracy is indicated in bold.

Features	OA (%)
RGB	47.55
LWIR	45.84
LWIR, RGB	54.38
LWIR, RGB, geometric	**57.78**
SWIR	73.77
SWIR, RGB	**73.85**
SWIR, RGB, geometric	72.08

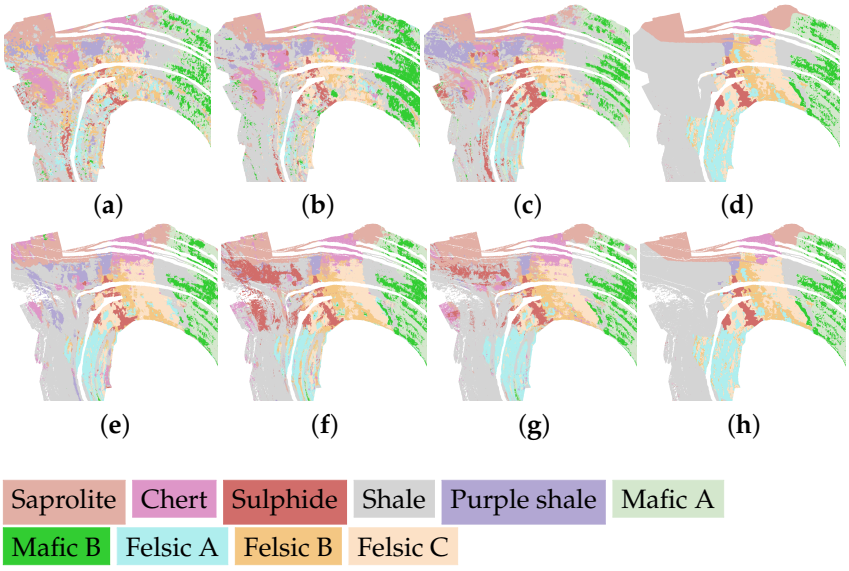

Figure 9. Different configuration of input features. (**a**) LWIR; (**b**) LWIR + RGB; (**c**) LWIR + RGB + geo; (**e**) SWIR; (**f**) SWIR + RGB; (**g**) SWIR + RGB + geo; (**d**,**h**) ground truth.

5.2. Multi-Stream Network

Although combining all different data modalities increases accuracy, we have investigated a more elegant fusing method to improve the results. While stacking all different features into one feature vector may seem like a simple solution, expanding the network into a multi-stream network is the most effective way to handle all different features. This approach yields better results, particularly on the LWIR dataset, where implementing the multi-stream network improves overall accuracy by 6.49 percentage points. A similar, but less significant, impact can also be observed on the SWIR dataset, where the overall accuracy increases by 3.37 percentage points. Table 7 compares the overall accuracy obtained by a one-stream network and a multi-stream network. In Figure 10, we notice the evidence where the multi-stream architecture reduces the mislabeled points, particularly in the Purple Shale class. Ultimately, separating different data modalities into distinct streams would be the optimal architecture for the dataset. We believe that enabling the network to learn various features independently enhances prediction results.

Table 7. The impact of implementing multi-stream network.

Features	Multi-Stream	OA (%)
LWIR, RGB, geometric	No	57.78
LWIR, RGB, geometric	Yes	64.27
SWIR, RGB, geometric	No	72.08
SWIR, RGB, geometric	Yes	75.45

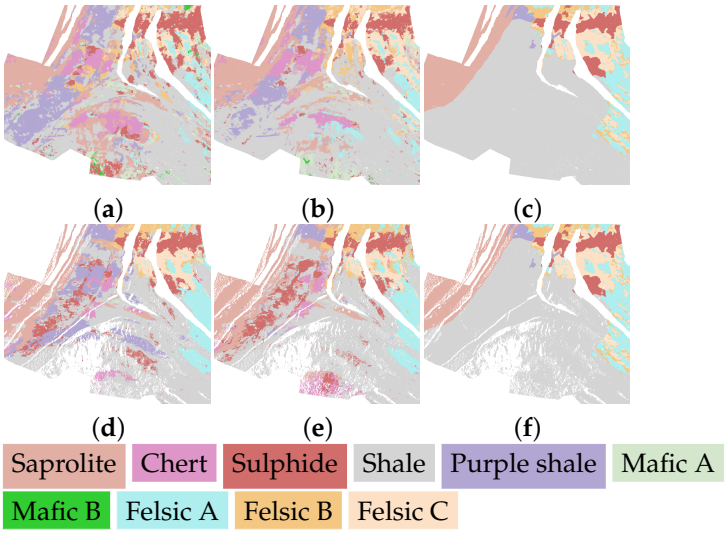

Figure 10. The impact of multi-stream (MS) network. (**a**) LWIR without MS; (**b**) LWIR with MS; (**d**) SWIR without MS; (**e**) SWIR with MS; (**c**,**f**) ground truth.

5.3. The 3D CNNs as Channel Attention Module

We added 3D convolutional filters following the graph reconstruction on the hyperspectral stream as the channel attention module to exploit the spectral information within the hyperspectral data. Table 8 shows our experiment results of the multi-stream network with the addition of 3D convolutional filters. The OA of the LWIR and SWIR datasets increased by 2.19 and 0.76 percentage points after employing the 3D convolutional filters. Again, we can witness the influence of adding 3D CNN by visually inspecting the predicted points in Figure 11. It is clear that after utilizing 3D CNN, the mislabeled classes Chert and Sulfide are then correctly predicted as Shale.

Table 8. The impact of adding 3D CNN in the hyperspectral stream.

Features	3D CNN	OA (%)
LWIR, RGB, geometric	No	64.27
LWIR, RGB, geometric	Yes	66.46
SWIR, RGB, geometric	No	75.45
SWIR, RGB, geometric	Yes	76.21

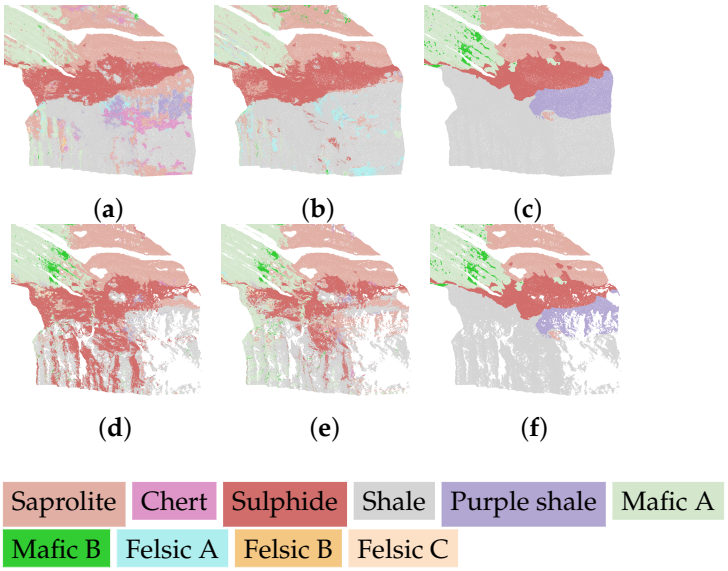

Figure 11. The impact of adding 3D CNN. (**a**) LWIR without 3DCNN; (**b**) LWIR with 3D CNN; (**d**) SWIR without 3DCNN; (**e**) SWIR with 3DCNN; (**c**,**f**) ground truth.

5.4. Geometric Features Transformation

We examined the significance of transforming the XYZ coordinates into the popular eigenvalues-based geometric features. The most primitive point cloud format only consists of XYZ coordinates, as that information is mandatory. Therefore, a point cloud is a data format to store 3D geometry information. In that sense, deep learning networks especially designed for point clouds would extract geometrical features from the XYZ coordinates. However, we faced a different situation here where we insisted that the spectral information of the dataset is a strong predictor for the segmentation, yet directly involving XYZ coordinates resulted in worse accuracy. Hence, omitting XYZ coordinates would be the straightforward solution. Rather than neglecting them, we propose to transform XYZ coordinates into eigenvalue-based features.

We compare the results from two different configurations. Firstly, we used XYZ coordinates and spectral features (hyperspectral and RGB) for the input data. Secondly, we transformed XYZ coordinates into geometric features and combined them with spectral features. We present the quantitative results in Table 9. Qualitatively, as Figure 12 suggests, we argue that our approach increases the true positive rate of the Sulfide class in the LWIR dataset (from 53.91% to 74.49%) and decreases the false positive rate of the same class in the SWIR dataset (from 7.72% to 5.47%).

Table 9. The impact of transforming XYZ coordinates into the eigenvalue-based geometric features.

Features	OA (%)
LWIR with XYZ	54.38
LWIR with geometric features	57.78
SWIR with XYZ	73.85
SWIR with geometric features	75.45

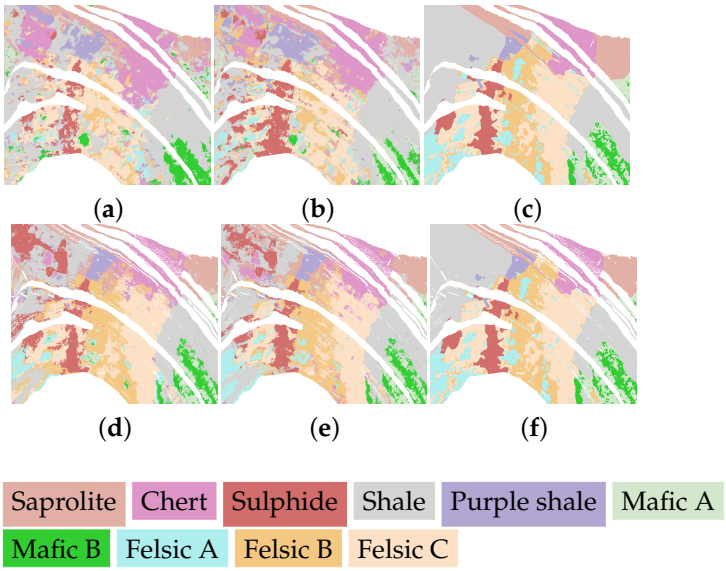

Figure 12. The impact of transforming XYZ into geometric features. (**a**) LWIR with XYZ; (**b**) LWIR with geometric features; (**d**) SWIR with XYZ; (**e**) SWIR with geometric features; (**c**,**f**) ground truth.

5.5. Choice of the Backbone

Finally, we examined the selection of the backbone for the multi-stream architecture. Specifically, we conducted a comparative analysis between multi-stream networks employing different encoders, namely EdgeConv and Self Attention (SA). The outcomes of this investigation are detailed in Table 10. The SA encoder is adopted from the global transformer layers as utilized in PCT [15]. Our experiments demonstrated that the EdgeConv-based multi-stream network outperformed the SA-based multi-stream network. Hence, we chose EdgeConv as the backbone of our proposed framework.

Table 10. The comparison of graph and transformer-based multi-stream network.

Backbone	OA (%)
LWIR with EdgeConv	64.27
LWIR with SA	54.65
SWIR with EdgeConv	75.45
SWIR with SA	65.80

6. Conclusions

This study presents a novel framework for mineral classification by employing multimodal point cloud data. Our approach involves a multi-stream architecture that combines data from different modalities, including XYZ, hyperspectral, and RGB. Our findings demonstrate that leveraging multimodal data yields superior performance compared to using unimodal data alone. Moreover, feeding distinct data modalities into separate streams enhances our model's performance. Additionally, we conducted an ablation study to gain insights into the significance of each component in our framework. Furthermore, we observed that point-based segmentation significantly outperforms image-based segmentation.

While the proposed framework has shown promising results, segmentation outcomes are subject to variations depending on the hyperspectral information. Specifically, the model faces challenges in effectively learning hyperspectral information for segmenting certain classes, such as Purple Shale in specific regions. In future research, we recommend exploring the fusion of multisensor hyperspectral data and developing a model capable of learning from diverse information sources. Additionally, we propose investigating various data fusion mechanisms and their optimal applications. Furthermore, we advocate for

assigning higher weighting to sensors that capture more valuable information, aiming to enhance segmentation accuracy.

To summarize, the evidence presented in this study highlights the potential of a multi-stream architecture for processing multimodal hyperspectral point cloud data. By further investigating the fusion of hyperspectral and point cloud data, we can create more robust segmentation models that are suitable for geological applications.

Author Contributions: Conceptualization, A.R. and P.G.; methodology, A.R. and A.J.A.; validation, A.R.; formal analysis, A.R.; investigation, A.R., A.J.A. and P.G.; writing—original draft preparation, A.R.; writing—review and editing, A.J.A., P.G. and R.G.; visualization, A.R.; supervision, P.G. and R.G.; project administration, R.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the European Regional Development Fund and the Land of Saxony by providing the high specification Nvidia A100 GPU server that we used in our experiments.

Data Availability Statement: The dataset is available at https://rodare.hzdr.de/record/2256 (accessed on 23 June 2024).

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

References

1. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]
2. Ben Hamida, A.; Benoit, A.; Lambert, P.; Ben-Amar, C. Deep Learning Approach for Remote Sensing Image Analysis. In Proceedings of the Big Data from Space (BiDS'16), Santa Cruz de Tenerife, Spain, 15–17 March 2016; Giorgio, S.P.M.P., Ed.; Publications Office of the European Union: Luxembourg, 2016; p. 133. [CrossRef]
3. Lee, H.; Kwon, H. Going Deeper With Contextual CNN for Hyperspectral Image Classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [CrossRef]
4. Li, Y.; Zhang, H.; Shen, Q. Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
5. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep Learning for Hyperspectral Image Classification: An Overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
6. Lorenz, S.; Salehi, S.; Kirsch, M.; Zimmermann, R.; Unger, G.; Vest Sørensen, E.; Gloaguen, R. Radiometric Correction and 3D Integration of Long-Range Ground-Based Hyperspectral Imagery for Mineral Exploration of Vertical Outcrops. *Remote Sens.* **2018**, *10*, 176. [CrossRef]
7. Kirsch, M.; Lorenz, S.; Zimmermann, R.; Tusa, L.; Möckel, R.; Hödl, P.; Booysen, R.; Khodadadzadeh, M.; Gloaguen, R. Integration of Terrestrial and Drone-Borne Hyperspectral and Photogrammetric Sensing Methods for Exploration Mapping and Mining Monitoring. *Remote Sens.* **2018**, *10*, 1366. [CrossRef]
8. Thiele, S.T.; Lorenz, S.; Kirsch, M.; Cecilia Contreras Acosta, I.; Tusa, L.; Herrmann, E.; Möckel, R.; Gloaguen, R. Multi-scale, multi-sensor data integration for automated 3-D geological mapping. *Ore Geol. Rev.* **2021**, *136*, 104252. [CrossRef]
9. Afifi, A.J.; Thiele, S.T.; Rizaldy, A.; Lorenz, S.; Ghamisi, P.; Tolosana-Delgado, R.; Kirsch, M.; Gloaguen, R.; Heizmann, M. Tinto: Multisensor Benchmark for 3-D Hyperspectral Point Cloud Segmentation in the Geosciences. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5501015. [CrossRef]
10. Ramachandram, D.; Taylor, G.W. Deep Multimodal Learning: A Survey on Recent Advances and Trends. *IEEE Signal Process. Mag.* **2017**, *34*, 96–108. [CrossRef]
11. Boulahia, S.; Amamra, A.; Madi, M. Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. *Mach. Vis. Appl.* **2021**, *32*, 121. [CrossRef]
12. Ma, X.; Zhang, X.; Pun, M.O. A Crossmodal Multiscale Fusion Network for Semantic Segmentation of Remote Sensing Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 3463–3474. [CrossRef]
13. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–10.
14. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph. (tog)* **2019**, *38*, 146. [CrossRef]
15. Guo, M.H.; Cai, J.X.; Liu, Z.N.; Mu, T.J.; Martin, R.R.; Hu, S.M. Pct: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [CrossRef]
16. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision—ECCV 2018, Proceedings of the 15th European Conference, Munich, Germany, 8–14 September 2018*; Proceedings, Part VII; Springer: Cham, Switzerland, 2018; pp. 3–19. [CrossRef]

17. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]
18. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual Attention Network for Scene Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3141–3149. [CrossRef]
19. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539. [CrossRef]
20. Ghamisi, P.; Maggiori, E.; Li, S.; Souza, R.; Tarablaka, Y.; Moser, G.; De Giorgi, A.; Fang, L.; Chen, Y.; Chi, M.; et al. New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning. *IEEE Geosci. Remote Sens. Mag.* **2018**, *6*, 10–43. [CrossRef]
21. Ghamisi, P.; Plaza, J.; Chen, Y.; Li, J.; Plaza, A.J. Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–32. [CrossRef]
22. Goel, P.; Prasher, S.; Patel, R.; Landry, J.; Bonnell, R.; Viau, A. Classification of hyperspectral data by decision trees and artificial neural networks to identify weed stress and nitrogen status of corn. *Comput. Electron. Agric.* **2003**, *39*, 67–93. [CrossRef]
23. Ratle, F.; Camps-Valls, G.; Weston, J. Semisupervised Neural Networks for Efficient Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 2271–2282. [CrossRef]
24. Mou, L.; Ghamisi, P.; Zhu, X.X. Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [CrossRef]
25. Wu, H.; Prasad, S. Semi-Supervised Deep Learning Using Pseudo Labels for Hyperspectral Image Classification. *IEEE Trans. Image Process.* **2018**, *27*, 1259–1270. [CrossRef] [PubMed]
26. Wu, H.; Prasad, S. Convolutional Recurrent Neural Networks for Hyperspectral Data Classification. *Remote Sens.* **2017**, *9*, 298. [CrossRef]
27. Ham, J.; Chen, Y.; Crawford, M.; Ghosh, J. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 492–501. [CrossRef]
28. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
29. Tarabalka, Y.; Chanussot, J.; Benediktsson, J. Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recognit.* **2010**, *43*, 2367–2379. [CrossRef]
30. Fauvel, M.; Tarabalka, Y.; Benediktsson, J.A.; Chanussot, J.; Tilton, J.C. Advances in Spectral-Spatial Classification of Hyperspectral Images. *Proc. IEEE* **2013**, *101*, 652–675. [CrossRef]
31. Aptoula, E.; Dalla Mura, M.; Lefèvre, S. Vector Attribute Profiles for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3208–3220. [CrossRef]
32. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; pp. 4959–4962. [CrossRef]
33. Slavkovikj, V.; Verstockt, S.; De Neve, W.; Van Hoecke, S.; Van de Walle, R. Hyperspectral Image Classification with Convolutional Neural Networks. In Proceedings of the 23rd ACM International Conference on Multimedia, New York, NY, USA, 26–30 October 2015; pp. 1159–1162. [CrossRef]
34. Zhao, W.; Du, S. Spectral–Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [CrossRef]
35. Yue, J.; Zhao, W.; Mao, S.; Liu, H. Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sens. Lett.* **2015**, *6*, 468–477. [CrossRef]
36. Audebert, N.; Saux, B.L.; Lefèvre, S. Deep Learning for Classification of Hyperspectral Data: A Comparative Review. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 159–173. [CrossRef]
37. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4338–4364. [CrossRef] [PubMed]
38. Bello, S.A.; Yu, S.; Wang, C.; Adam, J.M.; Li, J. Deep learning on 3D point clouds. *Remote Sens.* **2020**, *12*, 1729. [CrossRef]
39. Zhang, J.; Zhao, X.; Chen, Z.; Lu, Z. A Review of Deep Learning-Based Semantic Segmentation for Point Cloud. *IEEE Access* **2019**, *7*, 179118–179133. [CrossRef]
40. Xie, Y.; Tian, J.; Zhu, X.X. Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 38–59. [CrossRef]
41. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 945–953. [CrossRef]
42. Guerry, J.; Boulch, A.; Le Saux, B.; Moras, J.; Plyer, A.; Filliat, D. SnapNet-R: Consistent 3D Multi-view Semantic Labeling for Robotics. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 669–678. [CrossRef]

43. Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928. [CrossRef]
44. Le, T.; Duan, Y. PointGrid: A Deep Network for 3D Shape Understanding. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9204–9214. [CrossRef]
45. Graham, B.; Engelcke, M.; Maaten, L. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, 18–23 June 2018; pp. 9224–9232. [CrossRef]
46. Choy, C.; Gwak, J.; Savarese, S. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, 15–20 June 2019; pp. 3070–3079. [CrossRef]
47. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
48. Zhang, K.; Hao, M.; Wang, J.; Chen, X.; Leng, Y.; de Silva, C.W.; Fu, C. Linked Dynamic Graph CNN: Learning through Point Cloud by Linking Hierarchical Features. In Proceedings of the 2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Shanghai, China, 26–28 November 2021; pp. 7–12. [CrossRef]
49. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–11.
50. Boulch, A. ConvPoint: Continuous convolutions for point cloud processing. *Comput. Graph.* **2020**, *88*, 24–34. [CrossRef]
51. Thomas, H.; Qi, C.R.; Deschaud, J.; Marcotegui, B.; Goulette, F.; Guibas, L. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Los Alamitos, CA, USA, 27 October–2 November 2019; pp. 6410–6419. [CrossRef]
52. Wu, W.; Qi, Z.; Fuxin, L. PointConv: Deep Convolutional Networks on 3D Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9613–9622. [CrossRef]
53. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 16259–16268.
54. López, A.; Jurado, J.M.; Jiménez-Pérez, J.R.; Feito, F.R. Generation of hyperspectral point clouds: Mapping, compression and rendering. *Comput. Graph.* **2022**, *106*, 267–276. [CrossRef]
55. Brell, M.; Segl, K.; Guanter, L.; Bookhagen, B. 3D hyperspectral point cloud generation: Fusing airborne laser scanning and hyperspectral imaging sensors for improved object-based information extraction. *ISPRS J. Photogramm. Remote Sens.* **2019**, *149*, 200–214. [CrossRef]
56. Gader, P.; Zare, A.; Close, R.; Aitken, J.; Tuell, G. *Muufl Gulfport Hyperspectral and Lidar Airborne Data Set*; Technical Report REP-2013-570; University Florida: Gainesville, FL, USA, 2013.
57. Weinmann, M.; Weinmann, M. Fusion of hyperspectral, multispectral, color and 3d point cloud information for the semantic interpretation of urban environments. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 1899–1906. [CrossRef]
58. Weinmann, M.; Weinmann, M. Geospatial computer vision based on multi-modal data—How valuable is shape information for the extraction of semantic information? *Remote Sens.* **2017**, *10*, 2. [CrossRef]
59. Chen, B.; Shi, S.; Sun, J.; Gong, W.; Yang, J.; Du, L.; Guo, K.; Wang, B.; Chen, B. Hyperspectral lidar point cloud segmentation based on geometric and spectral information. *Opt. Express* **2019**, *27*, 24043–24059. [CrossRef] [PubMed]
60. Weidner, L.; Walton, G.; Krajnovich, A. Classifying rock slope materials in photogrammetric point clouds using robust color and geometric features. *ISPRS J. Photogramm. Remote Sens.* **2021**, *176*, 15–29. [CrossRef]
61. Nevalainen, O.; Honkavaara, E.; Tuominen, S.; Viljanen, N.; Hakala, T.; Yu, X.; Hyyppä, J.; Saari, H.; Pölönen, I.; Imai, N.N.; et al. Individual Tree Detection and Classification with UAV-Based Photogrammetric Point Clouds and Hyperspectral Imaging. *Remote Sens.* **2017**, *9*, 185. [CrossRef]
62. Decker, K.T.; Borghetti, B.J. Composite Style Pixel and Point Convolution-Based Deep Fusion Neural Network Architecture for the Semantic Segmentation of Hyperspectral and Lidar Data. *Remote Sens.* **2022**, *14*, 2113. [CrossRef]
63. Decker, K.T.; Borghetti, B.J. Hyperspectral Point Cloud Projection for the Semantic Segmentation of Multimodal Hyperspectral and Lidar Data with Point Convolution-Based Deep Fusion Neural Networks. *Appl. Sci.* **2023**, *13*, 8210. [CrossRef]
64. Xu, Y.; Du, B.; Zhang, L.; Cerra, D.; Pato, M.; Carmona, E.; Prasad, S.; Yokoya, N.; Hänsch, R.; Le Saux, B. Advanced Multi-Sensor Optical Remote Sensing for Urban Land Use and Land Cover Classification: Outcome of the 2018 IEEE GRSS Data Fusion Contest. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 1709–1724. [CrossRef]
65. Mitschke, I.; Wiemann, T.; Igelbrink, F.; Hertzberg, J. Hyperspectral 3D Point Cloud Segmentation Using RandLA-Net. In Proceedings of the International Conference on Intelligent Autonomous Systems, Zagreb, Croatia, 13–16 June 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 301–312.
66. Jaegle, A.; Borgeaud, S.; Alayrac, J.; Doersch, C.; Ionescu, C.; Ding, D.; Koppula, S.; Zoran, D.; Brock, A.; Shelhamer, E.; et al. Perceiver IO: A General Architecture for Structured Inputs & Outputs. In Proceedings of the Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, 25–29April 2022.

67. Li, P.; Gu, J.; Kuen, J.; Morariu, V.I.; Zhao, H.; Jain, R.; Manjunatha, V.; Liu, H. SelfDoc: Self-Supervised Document Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2021; pp. 5652–5660.
68. Hazirbas, C.; Ma, L.; Domokos, C.; Cremers, D. FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016.
69. Li, J.; Hong, D.; Gao, L.; Yao, J.; Zheng, K.; Zhang, B.; Chanussot, J. Deep learning in multimodal remote sensing data fusion: A comprehensive review. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102926. [CrossRef]
70. Chen, K.; Weinmann, M.; Gao, X.; Yan, M.; Hinz, S.; Jutzi, B.; Weinmann, M. Residual Shuffling Convolutional Neural Networks for Deep Semantic Image Segmentation Using Multi-Modal Data. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2018**, *IV-2*, 65–72. [CrossRef]
71. Chen, K.; Weinmann, M.; Sun, X.; Yan, M.; Hinz, S.; Jutzi, B.; Weinmann, M. Semantic Segmentation of Aerial Imagery via Multi-Scale Shuffling Convolutional Neural Networks with Deep Supervision. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2018**, *IV-1*, 29–36. [CrossRef]
72. Diakogiannis, F.I.; Waldner, F.; Caccetta, P.; Wu, C. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J. Photogramm. Remote Sens.* **2020**, *162*, 94–114. [CrossRef]
73. Audebert, N.; Le Saux, B.; Lefèvre, S. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 20–32. [CrossRef]
74. Kieu, N.; Nguyen, K.; Sridharan, S.; Fookes, C. General-Purpose Multimodal Transformer meets Remote Sensing Semantic Segmentation. *arXiv* **2023**, arXiv:2307.03388.
75. Concha, D.T.; Maia, H.D.A.; Pedrini, H.; Tacon, H.; Brito, A.D.S.; Chaves, H.D.L.; Vieira, M.B. Multi-stream convolutional neural networks for action recognition in video sequences based on adaptive visual rhythms. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 473–480.
76. Li, H.; Zech, J.; Hong, D.; Ghamisi, P.; Schultz, M.; Zipf, A. Leveraging openstreetmap and multimodal remote sensing data with joint deep learning for wastewater treatment plants detection. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *110*, 102804. [CrossRef] [PubMed]
77. Li, H.; Ghamisi, P.; Rasti, B.; Wu, Z.; Shapiro, A.; Schultz, M.; Zipf, A. A multi-sensor fusion framework based on coupled residual convolutional neural networks. *Remote Sens.* **2020**, *12*, 2067. [CrossRef]
78. Gao, J.; Li, P.; Chen, Z.; Zhang, J. A Survey on Deep Learning for Multimodal Data Fusion. *Neural Comput.* **2020**, *32*, 829–864. [CrossRef]
79. Baltrusaitis, T.; Ahuja, C.; Morency, L.P. Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 423–443. [CrossRef]
80. Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4558–4567.
81. Hackel, T.; Wegner, J.D.; Schindler, K. Contour detection in unstructured 3D point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1610–1618.
82. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the Importance of Initialization and Momentum in Deep Learning. In Proceedings of the 30th International Conference on International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; Volume 28, pp. 1139–1147.