University of Potsdam

Institute of Geosciences

DeepGeoMap:

A deep learning convolutional neural network architecture for geological hyperspectral classification and mapping

Master-Thesis submitted at
the Faculty of Science
for the degree

Master of Science

Master-Thesis by

Helge Leoard Carl Dämpfling

Published online on the
Publication Server of the University of Potsdam:
https://doi.org/10.25932/publishup-52057
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-520575

University of Potsdam

Institute of Geosciences

Remote Sensing, geoInformation and Visualization (RSIV)

Master-Thesis at
the Faculty of Science

DeepGeoMap:

A deep learning convolutional neural network architecture for geological hyperspectral classification and mapping

Candidate: Helge Dämpfling, B.Sc.

First examiner: Dr. Christian Mielke (GFZ Potsdam)

Second examiner: Prof. Dr. Uwe Altenberger (University of Potsdam)

June 2021

Abstract

In recent years, deep learning improved the way remote sensing data is processed. The classification of hyperspectral data is no exception. 2D or 3D convolutional neural networks have outperformed classical algorithms on hyperspectral image classification in many cases. However, geological hyperspectral image classification includes several challenges, often including spatially more complex objects than found in other disciplines of hyperspectral imaging that have more spatially similar objects (e.g., as in industrial applications, aerial urban- or farming land cover types). In geological hyperspectral image classification, classical algorithms that focus on the spectral domain still often show higher accuracy, more sensible results, or flexibility due to spatial information independence. In the framework of this thesis, inspired by classical machine learning algorithms that focus on the spectral domain like the binary feature fitting-(BFF) and the EnGeoMap algorithm, the author of this thesis proposes, develops, tests, and discusses a novel, spectrally focused, spatial information independent, deep multi-layer convolutional neural network, named 'DeepGeoMap', for hyperspectral geological data classification. More specifically, the architecture of DeepGeoMap uses a sequential series of different 1D convolutional neural networks layers and fully connected dense layers and utilizes rectified linear unit and softmax activation, 1D max and 1D global average pooling layers, additional dropout to prevent overfitting, and a categorical cross-entropy loss function with Adam gradient descent optimization. DeepGeoMap was realized using Python 3.7 and the machine and deep learning interface TensorFlow with graphical processing unit (GPU) acceleration. This 1D spectrally focused architecture allows DeepGeoMap models to be trained with hyperspectral laboratory image data of geochemically validated samples (e.g., ground truth samples for aerial or mine face images) and then use this laboratory trained model to classify other or larger scenes, similar to classical algorithms that use a spectral library of validated samples for image classification. The classification capabilities of DeepGeoMap have been tested using two geological hyperspectral image data sets. Both are geochemically validated hyperspectral data sets one based on iron ore and the other based on copper ore samples. The copper ore laboratory data set was used to train a DeepGeoMap model for the classification and analysis of a larger mine face scene within the Republic of Cyprus, where the samples originated from. Additionally, a benchmark satellite-based dataset, the Indian Pines data set, was used for training and testing. The classification accuracy of DeepGeoMap was compared to classical algorithms and other convolutional neural networks. It was shown that DeepGeoMap could achieve higher accuracies and outperform these classical algorithms and other neural networks in the geological hyperspectral image classification test cases. The spectral focus of DeepGeoMap was found to be the most considerable advantage compared to spectral-spatial classifiers like 2D or 3D neural networks. This enables DeepGeoMap models to train data independently of different spatial entities, shapes, and/or resolutions.

Zusammenfassung

In den letzten Jahren hat Deep Learning die Verarbeitung von Fernerkundungsdaten verbessert. Die Klassifizierung hyperspektraler Daten ist keine Ausnahme. 2D- oder 3D-Convolutional Neural Networks haben in vielen Fällen klassische Algorithmen zur hyperspektralen Bildklassifizierung übertroffen. Die Klassifikation geologischer hyperspektraler Bilder beinhaltet jedoch mehrere Herausforderungen, die oft räumlich komplexere Objekte umfassen als in anderen Disziplinen der hyperspektralen Bildanalyse, die in der Regel räumlich ähnlichere Objekte aufweisen (z. B. in industriellen Anwendungen, städtischen oder landwirtschaftlichen Luftaufnahmen). Bei der geologischen hyperspektralen Bildklassifizierung zeigen klassische Algorithmen, die sich auf den Spektralbereich konzentrieren, oft noch eine höhere Klassifizierungsgenauigkeit, sinnvollere Ergebnisse oder Flexibilität aufgrund räumlicher Informationsunabhängigkeit. Im Rahmen dieser Arbeit, inspiriert von klassischen maschinellen Lernalgorithmen, die sich auf den spektralen Bereich konzentrieren, wie dem Binary Feature Fitting-(BFF) und dem EnGeoMap-Algorithmus, schlägt der Autor dieser Arbeit ein neuartiges, spektral fokussiertes, räumlich unabhängiges, tiefes mehrschichtiges neuronales Faltungsnetzwerk (Deep Convolutional Neural Network) mit dem Namen "DeepGeoMap" für die hyperspektrale geologische Datenklassifizierung vor. Genauer gesagt verwendet die Architektur von DeepGeoMap eine sequenzielle Reihe verschiedener „1D-Convolutional-Layer“ und „1D-Dense-Layer“ und verwendet ReLU und Softmax-Aktivierung, "1D-Max- und 1D-Global-Average-Pooling-Layer“, ein zusätzliches "Dropout-Layer", um ein „Overfitting“ zu verhindern, und eine kategoriale Kreuzentropieverlustfunktion mit Adam-Gradientenabstiegsoptimierung. DeepGeoMap wurde mit Python 3.7 und der Machine- und Deep-Learning-Schnittstelle TensorFlow mit Grafikartenbeschleunigung (GPU) realisiert. Diese spektral fokussierte 1D-Architektur ermöglicht das Trainieren von DeepGeoMap-Modellen mit hyperspektralen Laborbilddaten geochemisch validierter Proben (nach dem Vorbild klassischer Algorithmen, die eine Spektralbibliothek validierter Proben zur Bildklassifizierung verwenden). Die Klassifizierungsfähigkeiten von DeepGeoMap wurden mit zwei geologischen hyperspektralen Bilddatensätzen getestet. Bei beiden handelt es sich um geochemisch validierte hyperspektrale Datensätze, von denen einer auf Eisenerz und der andere auf Kupfererzproben basiert. Der Kupfererz-Labordatensatz wurde verwendet, um ein DeepGeoMap-Modell für die Klassifizierung und Analyse einer größeren Tagebauwandszene in der Republik Zypern, aus der die Proben stammten, zu trainieren. Darüber hinaus wurde ein satellitenbasierter Benchmark-Datensatz, der Indian Pines-Datensatz, für Training und Tests verwendet. Die Klassifikationsgenauigkeit von DeepGeoMap wurde mit klassischen Algorithmen und anderen neuronalen Faltungsnetzen verglichen. Es wurde gezeigt, dass DeepGeoMap höhere Genauigkeiten erreichen und diese klassischen Algorithmen und andere neuronale Netze in den Testfällen der geologischen hyperspektralen Bildklassifizierung übertreffen kann. Der spektrale Fokus von DeepGeoMap erwies sich als der größte Vorteil gegenüber spektral-räumlichen Klassifikatoren wie 2D- oder 3D-Convolutional Neural Networks. Dadurch können DeepGeoMap-Modelle Daten unabhängig von unterschiedlichen räumlichen Einheiten, Formen und/oder Auflösungen trainieren.

Contents

Abbreviations

AA	Average Accuracy
ADALINE	ADAptive LINear Element
ANN	Artificial neural networks
API	Application Programming Interface
AVIRIS	Airborne Visible / Infrared Imaging Spectrometer
BIF	Banded Iron Formations
BFF	Binary Feature Fitting
CE	Cross-Entropy
CNN	Convolutional Neural Network
CPU	Central Processing Unit
GFZ	German Research Centre for Geosciences
EnGeoMAP	Automated Hyperspectral Mineral Identification for the German EnMAP Space Mission
FOV	Field of View
GPU	Graphics Processing Unit
GSD	Geological Survey Department of the Republic of Cyprus
HSI	Hyperspectral Image
HySpex	NEO's hyperspectral cameras
JSRC	Joint Sparse Representation Classification
LSMSC	Localized Spectral features and Multi-scale Spatial features Convolution
ML	Machine Learning
MLP	Multi-Layer Perceptron
MNIST	Modified National Institute of Standards and Technology
MoU	Memorandum of Understanding
NASA	National Aeronautics and Space Administration
NN	Neural Networks
OA	Overall Accuracy
PCA	Principal Components Analysis
REE	Rare Earth Elements
ReLU	Rectified Linear Unit
RGB	Red-Green-Blue
ROI	Region of Interest
SAM	Spectral Angle Mapper
SVM	Spectral Vector Machine
SWIR	ShortWave InfraRed
VA	Validation Accuracy
VL	Validation Loss
VMS	Volcanogenic Massive Sulfides
VNIR	Visible and Near InfraRed
QF	Quadrilátero Ferrífero

Ich versichere, dass ich die von mir vorgelegte Arbeit selbstständig verfasst habe, dass ich die verwendeten Quellen, Internet-Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Berlin, den 21. Juni 2021

. . . . . .

(Unterschrift des Kandidaten)

Motivation

The worldwide demand for critical mineral resources is at an all time high and growing as the world is becoming more industrialised and population growth accelerates this increase even further [9, 38]. The majority of the industry critical mineral resources are produced in developing countries in addition to the production from developed mining nations such as Canada, Australia, China and Russia. The group of industry critical materials comprise of copper, gold, iron, lead, zinc, and rare-earth minerals (REEs) [9, 38]. Developing industry nations as well as the highly industrialised world such as the European Union face new challenges through climate change [9]. The acknowledgment of the climate impact and the will to counter it and adapt greener industries, technologies and energy sources will create an additional demand towards ores rich in iron, copper, aluminium and REE's [9]. Only a small portion of the future resource demand can be generated by recycling today, thus demanding a further increase in mineral exploration and mining development for critical resources [15]. In the recent past, experts have conducted many scientific studies demonstrating the effectiveness of hyperspectral proximal and remote sensing image analysis with respect to geology and mineral exploration [14, 24, 53]. The key components of hyperspectral remote and proximal sensing are the hyperspectral cameras and the software to analyse and classify the image data acquired by the cameras [14, 24].

Lately, advances in computational power lead to the more widespread development and adoption of machine learning algorithms that utilise neural networks for hyperspectral image analysis and classification. Especially computationally demanding convolutional neural networks (CNNs), that were already shown to outperform less demanding simpler artificial neural networks (ANNs) in image classification in 1998 by LeCunn et al. (1998), suddenly gained popularity when Chellapilla et al. (2006) used a graphical processing unit to accelerate their training by up to four times.

In November 2015 Google published a white paper and code introducing a software interface named TensorFlow. TensorFlow aims to simplify the development of deep learning algorithms like convolutional neural networks for various purposes [36].

Hence the question arises: Can TensorFlow be utilized to develop a deep convolutional neural network for geological hyperspectral image classification and mapping from scratch? Here, a spectrally focused, spatial information independent, deep multilayer convolutional neural network is proposed, realised, tested, and discussed.

Goal of this thesis

The goal of this Master-Thesis is the development of a deep learning, convolutional neural network architecture using the programming language Python, and the ma-

chine learning interface TensorFlow that identifies rocks, minerals, or in general land cover types in hyperspectral images and creates classification maps of these, similar to the EnGeoMAP or Binary Feature Fitting (BFF) algorithms developed by Christian Mielke at the German Research Centre for Geosciences (GFZ) (see [38, 39, 24]).

1 Introduction

In the following, relevant concepts, terminology and computational frameworks are shortly introduced before the methods are presented in the next Section.

1.1 Hyperspectral Proximal and Remote Sensing

In the recent years hyperspectral imaging is becoming an ever more important field in remote and proximal sensing. This is caused by the fast development of new sensors which have increased greatly in terms of spatial and spectral resolution. Hyperspectral imaging sensors enable the detection of vastly more substances in comparison with multispectral imaging sensor types. In the geosciences, hyperspectral imaging can be used in many different areas such as the detection of geohazards, hydrogeology, mineral detection and exploration and many more (see [14, 24, 27, 38, 43, 52]).

Spectroscopy: The following introduction to Spectroscopy is based on Eismann (2012). Spectroscopy in general is based on the interaction of electromagnetic radiation with the material, which is the physical basis of hyperspectral imaging in terms of the direct relationship between observable material spectral features and the inherent compositional characteristics. It relates to electromagnetic interaction at the atomic and molecular level and requires a strictly quantum mechanical investigation, which is, however, limited to simple molecules due to the complexity of the quantum theory. Thus, the identification of materials composed of very complex mixtures of molecules usually relies on empirical relationships of apparent spectral properties, such as reflectance and emissivity spectra. The spectral properties are unique to a material and therefore be considered to provide a signature from which the material type, e.g. a mineral, can be determined [14].

Spectroscopy is not limited to crystalline materials and be used for identification of amorphous materials as well [42, 14]. It is is sensitive to the chemical bonds in the material and the crystalline structure and the texture of the surface [42, 14]. Simpler, minerals can be classified by the position and shape of their characteristic absorption features and semi-quantified by the depth of these features [42, 24]. Differences in chemical compositions lead to position shifts and shape changes of the spectral features which make them complex. Through identification of these changes and shifts, changes

of chemistry in the rock, and also more exactly proxies for elements and substitution of elements in a mineral can be determined[42, 14].

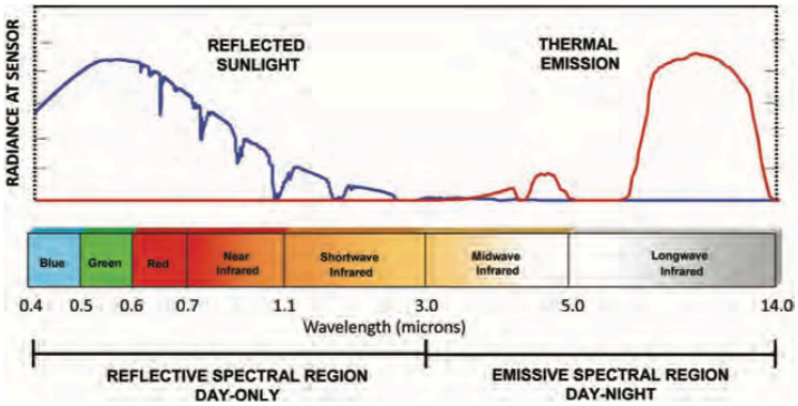

Figure 1: The main natural sources of earth observation and infrared radiation and the definition of specific spectral regions, Image:[14].

Hyperspectral imaging The following introduction to hyperspectral imaging is largely based on and Eismmann (2021). Hyperspectral remote sensing is the combination of two sensing methods familiar to most scientists and engineers: imaging and spectrometry. An imaging system (a hyperspectral camera) captures a picture of a scene related to the spatial distribution of the power of reflected and/or emitted electromagnetic radiation integrated over a spectral band. The most commonly used spectral regions are the visible and near infrared (VNIR) and the shortwave infrared regions (SWIR). Spectrometry measures the variation in power with the wavelength or frequency of light, capturing information related to the chemical composition of the materials measured. The instrumentation that is used to capture the spectral information is called a spectrometer . These two are combined in a spectral imaging sensor. "The physical basis of image content in VNIR imagery is a combination of broadband reflectance differences between objects (shape) and illumination variation due to the 3D nature of the scene (shading)", Eismann (2012). This broadband imaging concept can be extended to multispectral imaging, where simultaneously images are captured in multiple spectral bands where the pixels are spatially registered, so that they view identical parts of the scene. Multispectral imagery can support image classification or thematic mapping into regions based on the spectral (and hence material) content, or object detection based on its underlying spectral characteristics. Multispectral analysis, however is limited, to the spectral information captured by the small and discrete number of spectral bands. Hyperspectral imaging systems on the other hand are able to simultaneously capture both the spatial and spectral content of scenes with excellent spatial and spectral resolution and coverage. The resulting data product can be called a hypercube which is a 3D dataset composed of layers of grayscale images, each

pixel of which contains a finely sampled spectrum whose features are related to the materials contained within it (figure 2). A spectrum can simply be defined as the distribution of some measure of light power or material property with optical wavelength, such as the radiance spectrum. "The spectrum contains information related to the chemical composition of the material composing the spatial pixel of the imaging spectrometer, information that arises through physical relationships between the spectrum of reflected or emitted light, the vibrational and electronic resonances of molecules composing the material, and microscopic surface and volumetric properties", Eismann (2012). What distinguishes a hyperspectral sensor from a multispectral sensor is that a hyperspectral imaging sensor has the ability to capture a continuous spectrum as opposed to discrete sets of often discontinuous spectral bands in a multispectral sensor [14].

Another term, 'superspectral imaging' is loosely describing imagery with band numbers between the multi- and hyperspectral resolution (between 15 and 100 bands) (figure 3) [24]. Superspectral imaging can be used to distinguish broad spectral features (e.g. iron absorptions) but no narrow features that are not resolved [24]. This thesis will focus on hyperspectral imaging classification, but will consider multi- and or superspectral resolution analysis from other sources for comparison.

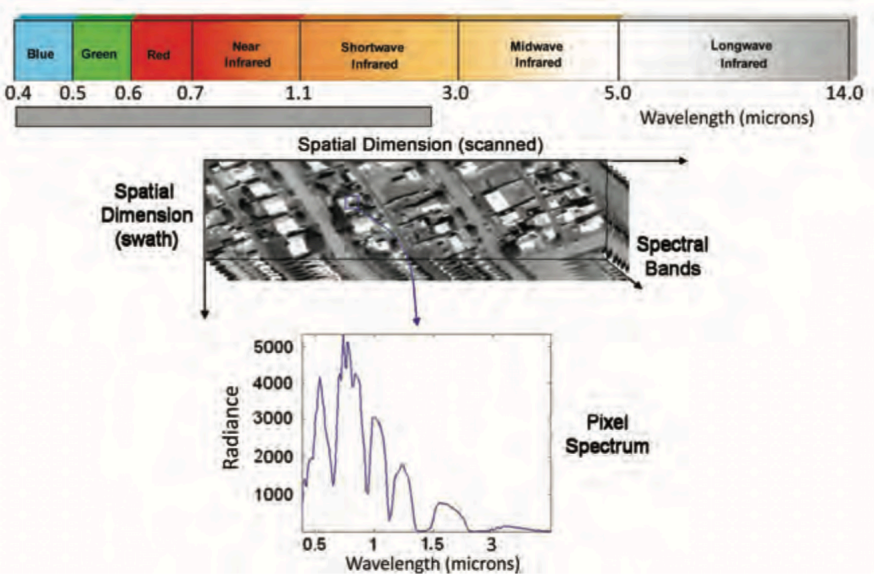

Figure 2: "Illustrative examples of hyperspectral imagery across the VNIR/SWIR spectral region. Finely sampled spectral bands across the continuous spectral region can be assembled into a 3D hypercube, where a trace in the spectral direction for each pixel contains a spectrum related to the pixel material content.", Eismann (2012), Image:[14].

Remote sensing in terms of hyperspectral imaging usually involves the measurement by aircraft or satellite on large scales, but hyperspectral imaging cameras can also be utilized in the laboratory on smaller scale objects like rock samples, which then can be referred to as proximal sensing. Hyperspectral imaging cameras have been successfully

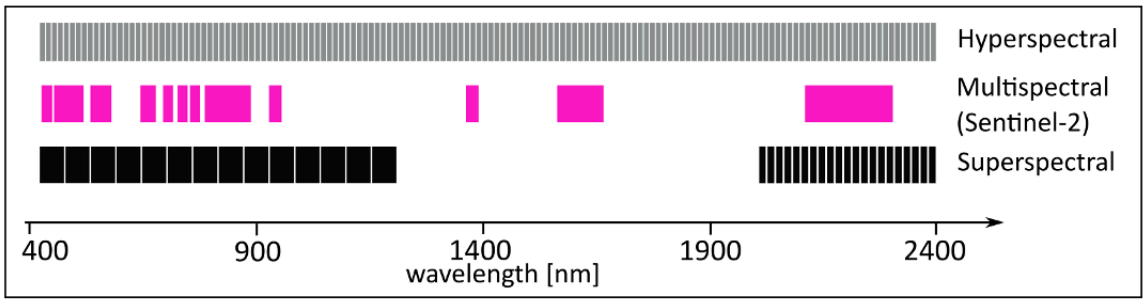

Figure 3: "Schematic visualization of band width and number differences between hyper-, super- and multispectral sensors.", Koerting (2021), Image:[24].

used to detect minerals and elements in outcrops, rock samples, and drill cores using reflectance measurements. The detection and classification of minerals and elements spaceborne, airborne and in the laboratory is crucial in both finding and evaluating outcrops and ore deposits for exploration and in scientific research (see [38, 39, 26, 24]).

1.2 Deep Learning, Artificial Intelligence, Neural Networks, TensorFlow

Deep Learning

Deep learning is a certain kind of machine learning that can achieve high accuracy and flexibility and solves a central problem in representation learning[18]. Deep learning is introducing representations that are expressed in terms of other, simpler representations, and hence allows the computer to build complex concepts out of simpler concepts [18].

The prototypical example of a deep learning model is the feedforward deep network or multilayer perceptron (MLP)[18]. A multilayer perceptron is a mathematical function mapping a set of input values to output values [18].

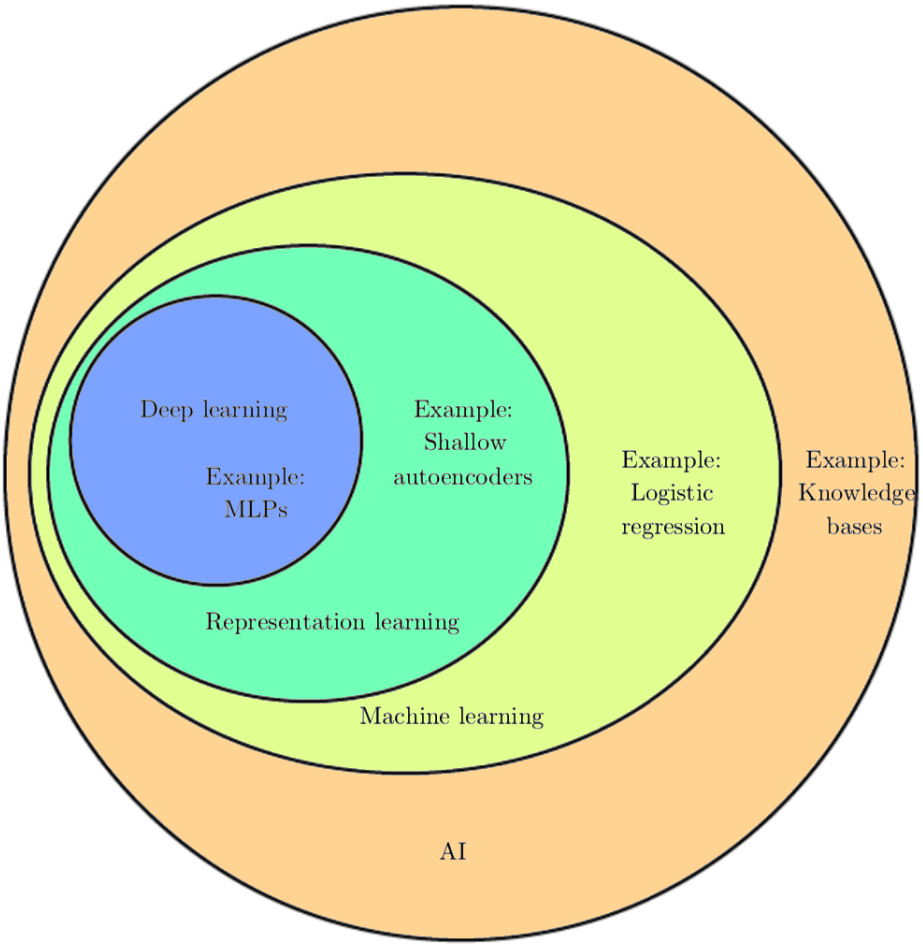

Figure 4: Venn diagram indication how deep learning is a kind of representation learning, which is part of the larger realm of machine learning, which part of Artificial Intelligence (AI). Each section includes a technological example. Image modified after: Goodfellow et al.(2016)

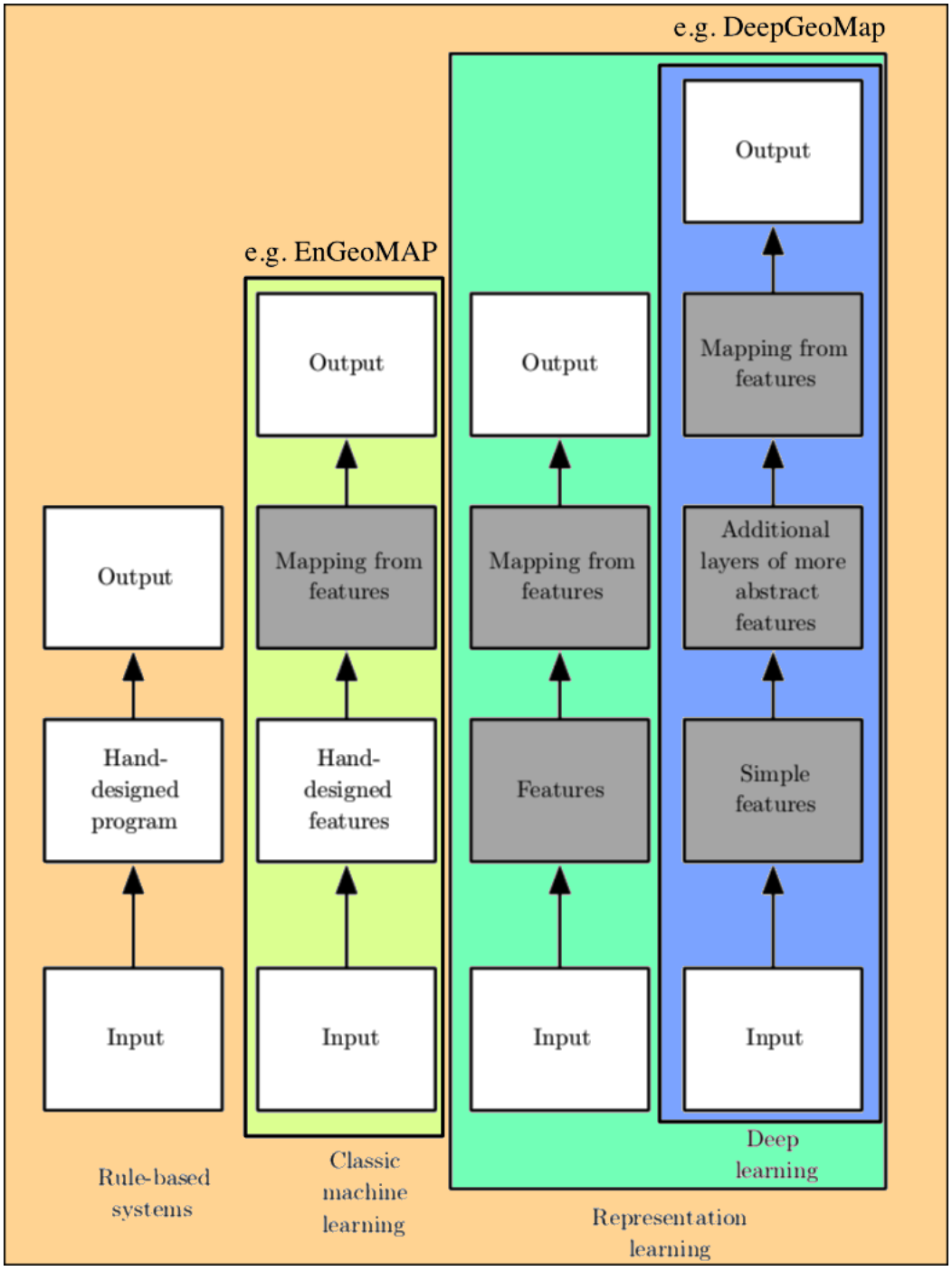

Figure 5: Flowcharts depicting how the different subsections AI relate to each other within different AI disciplines. Image modified after: Goodfellow et al.(2016)

1.2.1 Neural Networks

Artificial neurons used in conventional artificial neural networks (ANNs), or simply neural networks (NNs), are the first-order (linear) models of biological neurons [22]. The following description of the biological basis of the artificial neuron is based on Kiranyaz et al. (2019). In the biological neural systems, the learning is mainly performed at the cellular level. Figure 6, shows a biological neuron. Each neuron in a biological brain is capable of processing the electrical signal based on the three individual operations: "1) reception of the other neurons outputs through the synaptic connections in Dendrites, 2) the integration (or pooling) of the processed output signals in the soma at the nucleus of the cell, and, 3) the activation of the final signal at the first part of the Axon or the so-called Axon hillock: if the pooled potentials exceed a certain limit, it "activates" a series of pulses (action potentials)", Kiranyaz (2019).

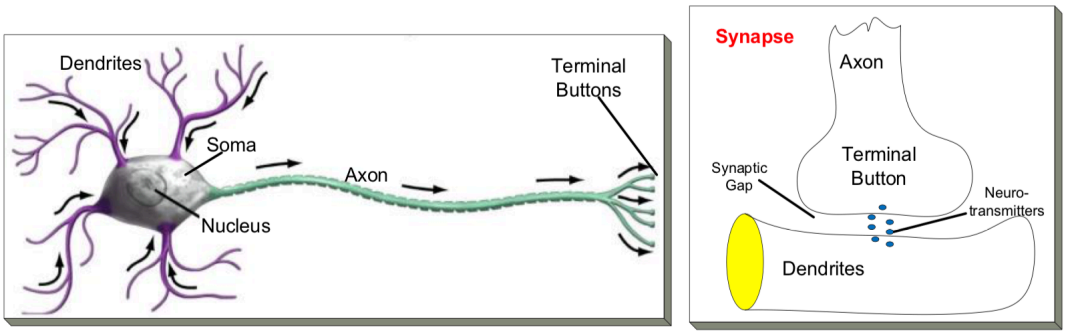

Figure 6: On the left side a biological neuron indicating the direction of the signal flow, and on the right side a synapse is shown. Image:[23]

Figure 6, shows that each terminal button is connected to other neurons across a small gap called a synapse [22]. During the 1940s the first "artificial neuron" model was proposed by McCulloch-Pitts, which performs a linear transformation through a weighted summation by the scalar weights. Hence, "the basic operations performed in a biological neuron, that operate the individual synaptic connections with specific neurochemical operations and the integration in the cell's soma are modeled as the linear transformation (linear weighted sum) followed by a possibly nonlinear thresholding function, which is called **activation function**.", Kiranyaz (2019). Hence the McCulloch-Pitts Neuron is an early model of brain function [22].

The following introduction the history of neural networks is mainly based on Goodfellow et al. (2016). These first models were designed to take a set of n input values $x_1, ..., x_n$ and correlate them with an output y. A set of **weights** would be learned $w_1, ..., w_n$ and the output computed: $f(\boldsymbol{x}, \boldsymbol{w}) = x_1 w_1 + \cdots + x_n w_n$ [18]. These linear models could recognise two different categories of inputs by testing whether $f(x, w)$ is positive or negative. For the model to correspond to the desired definition of the categories, the weights needed to be set correctly by a human operator . Then the

so called "perceptron" (Rosenblatt, 1958, 1962) became the first model that could learn the weights defining the categories given examples of inputs from each category. An additional method the adaptive linear element (ADALINE) by Widrow and Hoff (1960), returned the value of $f(x)$ itself to predict a real number, and can also learn to predict these numbers from data. The training algorithm used to adapt the weights of the ADALINE is an algorithm called **stochastic gradient descent** [18]. Modified versions of the stochastic gradient descent algorithm are still the dominant cost (or loss) functions for deep learning models today. "Models based on the $f(x, w)$ used by the perceptron and ADALINE are called linear models.", Goodfellow (2016). Models based on this remain some of the most used machine learning models, although they are often trained in different ways than the original models were trained, for example with back-propagation. Linear models have many limitations, including that they cannot learn the XOR function [18].

Most neural networks today are based on a model neuron called the **rectified linear unit (ReLU)** (see section 2.1.1 [18, 28]. The idea of having many computational units become intelligent only via their interactions with each other is a concept inspired by the brain and models can be called **multilayer perceptron (MLP)** [18]. All these accomplishments have started a new wave of approaches that eventually lead to the first **Convolutional Neural Network (CNN)** models[18, 28]. But CNNs require a lot of computational resources and only advancements in computational power made them a feasible concept [18, 28]. In 1990 Yann LeCun formulated the **back-propagation** to train the first CNN, the so- called "LeNet" [31]. Modern networks use this back-propagation to adjust the connection weights to compensate for errors during learning (training), where the error amount is effectively divided among the connections, and hence the gradient (or derivative) of the cost (loss) function is calculated [28]. The superior classification capabilities of a CNN was first demonstrated over the benchmark MNIST handwritten number database by Yann LeCun in 1998 (see [31]).

1.2.2 TensorFlow

TensorFlow is a computer programming interface for expressing machine learning algorithms, and an implementation for the execution of such algorithms introduced and released by Google to the public in November 2015, the following descriptions are based on this original publication by Google (see [36]).

Computations expressed using TensorFlow can be executed with little to no change on a large variety of heterogeneous systems, ranging from mobile devices such as phones and tablets up to large-scale distributed systems of hundreds of machines and thousands of computational devices such as GPU cards. TensorFlow is a flexible system

that can be used to express a wide variety of algorithms, including training and inference algorithms for deep neural network models. It has been applied throughout science, computer science and other fields, with projects including speech recognition, computer vision, robotics, information retrieval, natural language processing, geographic information extraction, and computational drug discovery. The TensorFlow computations are expressed as stateful dataflow graphs, and are focused to make the system both flexible enough for quick experimentation with new models for research purposes and sufficiently high performance and robust for production training and deployment of machine learning models. For scaling neural network training to larger deployments, TensorFlow allows to easily express various kinds of parallelism through replication and parallel execution of a core model dataflow graph, with many different computational devices all collaborating to update a set of shared parameters or other state [36] .

A computational graph is typically constructed using one of the supported frontend languages (C++ or Python) [36]. In this thesis Python 3.7 is used (see Section 2). An example fragment to construct and then execute a TensorFlow graph using the Python front end is shown in Figure 7, and the resulting computation graph in Figure 8 [36].

```
import tensorflow as tf

b = tf.Variable(tf.zeros([100]))                    # 100-d vector, init to zeroes
W = tf.Variable(tf.random_uniform([784,100],-1,1)) # 784x100 matrix w/rnd vals
x = tf.placeholder(name="x")                        # Placeholder for input
relu = tf.nn.relu(tf.matmul(W, x) + b)              # Relu(Wx+b)
C = [...]                                           # Cost computed as a function
                                                    # of Relu


s = tf.Session()
for step in xrange(0, 10):
  input = ...construct 100-D input array ...        # Create 100-d vector for input
  result = s.run(C, feed_dict={x: input})           # Fetch cost, feeding x=input
  print step, result
```

Figure 7: Examplary TensorFlow code fragment [36].

In a TensorFlow graph, the instantiation of an **operation** is represented by a **node** and each node has zero or more inputs and zero or more outputs. Values that flow along normal edges from outputs to inputs in the graph are **tensors**, arbitrary dimensionality arrays where the underlying element type is specified or inferred at graph-construction time. **Control dependencies** can also exist in the graph. These Control dependencies are special edges where no data flows along, but they indicate that the source node for the control dependence must finish executing before the destination node for the control dependence starts executing. Since TensorFlow includes mutable state, control dependencies can be used directly by **clients** to enforce *happens* before *relationships*. In computing the term client, refers to a piece of computer hard-

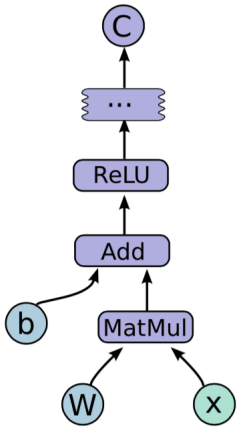

Figure 8: Corresponding computation graph for the code including the variables in Figure 7 [36].

ware or in this case software that accesses a service (in this case TensorFlow) made available by a server or other computer hardware (a computational **device**, such as a CPU or GPU). The TensorFlow implementation also sometimes inserts control dependencies to enforce *orderings* between otherwise independent operations as a way of, for example, controlling the peak memory usage. The following paragraphs will explain some important features and terminology of TensorFlow [36].

Tensors: A tensor in TensorFlow is a typed, multi-dimensional array. A variety of tensor element types are supported in TensoFlow, including signed and unsigned integers ranging in size from 8 bits to 64 bits, IEEE float and double types, a complex number type, and a string type (an arbitrary byte array) [36].

Operations: In TensorFlow an operation has a name and represents an abstract computation (e.g., "matrix multiply", or "add"). Such an operation can have **attributes**, which must be provided or inferred at graph-construction time in order to instantiate a node to perform the operation. A frequent use of attributes is making operations polymorphic over different tensor element types for example, *add* of two tensors of type float versus *add* of two tensors of type int32 [36].

Devices: Devices are the computational core element of TensorFlow, where each worker (e.g. a Server or Computer) is responsible for one or more devices (e.g. a CPU, or a GPU, or multiple GPU's), and each device has a device type (e.g. CPU), and a name (e.g. "/job:localhost/device:cpu:0"). The device names are constructed from pieces that identify the device's type, the device's index within the worker, and, in our distributed setting, an identification of the job and task of the worker (or localhost for the case where the devices are local to the process) [36].

Kernels: A kernel is a certain implementation of an operation that can be run on a particular type of device like a CPU or a GPU. A TensorFlow binary defines the sets of operations and kernels available via a registration mechanism . TensorFlow binaries can be extended by linking in additional operation and/or kernel definitions/registrations to them [36].

Sessions: Sessions are the way clients programs interact with the TensorFlow system. The initial computation graph when a session is created is empty. The Session interface therefore supports an *Extend* method in order to augment the current graph managed by the session with additional nodes and edges. The second major operation supported by the session interface is *Run*. The *Run* operation takes a set of output names that need to be computed, additional to an optional set of tensors to be fed into the graph in place of certain outputs of nodes. With the arguments to *Run*, the TensorFlow implementation can compute the transitive closure of all nodes that must be executed in order to compute the outputs that were requested. TensorFlow can then arrange to execute the appropriate nodes in an order that respects their dependencies [36].

Variables: In most TensorFlow computations a graph is executed multiple times, where most tensors do not survive past a single execution of the graph. A special kind of operation that returns a handle to a persistent mutable tensor and survives across executions of a graph is called a Variable. Handles to these persistent mutable tensors can be passed to special operations, such as Assign and AssignAdd (representing +=) which mutate the referenced tensor. In TensorFlow machine learning applications, the parameters of the model are typically stored in tensors held in variables, in which these are updated as part of the Run of the training graph for the model [36].

Device execution: TensorFlow is capable of both single- and multi-device execution. For this thesis only the single worker process with a single device is considered as the whole development took place on a computer with a single graphics card device (see Section 2). In single device execution the nodes of the graph are executed in an order that respects the dependencies between nodes [36].

The TensorFlow API and a reference implementation were released as an open-source package under the Apache 2.0 license in November, 2015 and are available at www.tensorflow.org [36].

For a more detailed description of TensorFlow and the detailed mechanics behind the functions described above please consider the original white paper [36].

2 Methods

The proposed architecture was developed using Python version 3.7 and TensorFlow version 2.3.1. The samples and data are geochemically verified and published, and/or a well known benchmark dataset. First the deep convolutional neural network architecture of DeepGeoMap is proposed (section 2.1), then the background (including geology, data-acquisition, and -processing) of the samples and data used in this thesis is presented (section 2.2), then the data pre-processing and preparation necessary for DeepGeoMap is explained (section 2.3).

2.1 DeepGeoMap: The proposed deep convolutional neural network

In the past decade there has been made tremendous progress in the field of neural networks, regarding hyperspectral image classification to point that they outperform classical algorithms in accuracy for many hyperspectral data sets (see [1, 30, 32, 40, 50, 51]). Most of these neural architectures use spectral-spatial information relationships of the hyperspectral data to achieve very high accuracies (over 95%) with very little training data (some use just 10% of the available training data). While this is state of the art and works very well on identifying clearly structured objects that can be resolved well using the right resolutions e.g. landscapes influenced by human activity including crop fields, infrastructure, urban areas, or on smaller scale biological objects such as vegetation (plants, trees, fruits, etc.) it can be assumed that the spectral-spatial advantage will decrease the more diversely structured, chaotically structured or structurally less resolved the objects of interest are (e.g. sand grains, soil, small rocks, certain minerals of rocks in larger scale images, etc.). Furthermore the more the neural networks accuracy is bound to spatial information of the pixels the less flexible a pre-trained model will be regarding other images of different scale, and or resolution. With some exceptions the broad fields of geological and soil remote and proximal sensing face the challenges of involving much more unstructured or much more differently, independently structured objects of interest. Hence an architecture for geological hyperspectral imaging, named DeepGeoMap, is proposed that solely focuses on the spectral information available. This allows the models of the proposed neural architecture to be trained with spectral data from many different instruments including point-spectrometers. DeepGeoMaps architecture represents essentially a signal analysis tool and can be used for any kind of 1 dimensional (1D) signal data classification and categorisation. Additionally, one of the main ideas is to be able to train the models with hyperspectral images of geochemically validated samples acquired under laboratory conditions (see section 2.2.3 & 2.2.1) to avoid or reduce the number

of false positives due to mixed-spectra or falsely labeled training data/pixels in lower resolution airborne or satellite data. A key aspect of this is therefore for also adequate pre-processing and selection of the laboratory acquired training data (see section 2.3). DeepGeoMap consists of sequential series of different 1D convolutional neural networks layers, and fully connected dense layers and utilises, rectified linear unit and softmax activation, 1D max pooling, 1D global average pooling layers, additional dropout to prevent overfitting and a categorical crossentropy loss function with an Adam gradient descent optimization. First a short overview of the general structure of DeepGeoMap is given, then the different parts and functions of the network are explained.

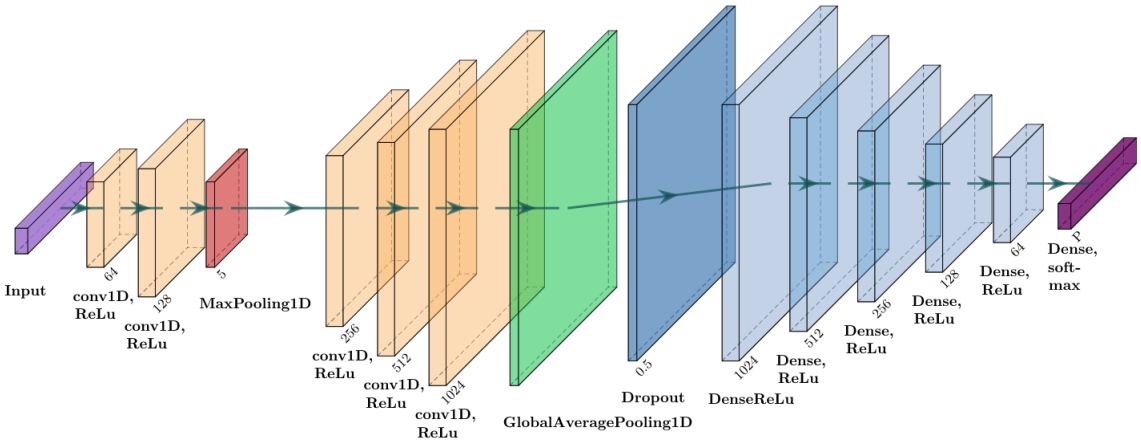

Figure 9: The generalised structure of a DeepGeoMap model.

DeepGeoMap is primarily targeted at hyperspectral image data classification that requires a dimensionality change and reduction by array flattening, but by design its models can also be trained with labeled hyperspectral point spectrometer data or used for completely different signal analysis, hence the data extraction and dimensionality reduction by array flattening is considered part of the data pre-processing (section 2.3).

DeepGeoMap consists of 5 1D convolutional neural layers with 'Rectified Linear Unit' (ReLU) activation, a 'max pooling 1D' layer, a 'global average pooling 1D' layer, a dropout layer, 5 fully connected dense layers (Multi layer perceptron; MLP) with ReLU activation, and a final fully connected 'dense' layer with softmax activation (figure 9). The kernel size of the convolutional layers is set to 3, while the filter size of the convolutional layers is increasing with each layer by a factor of 2. Beginning with a filter size of 64 and ending with 1024. Following the first 2 convolutional layers a 1D max pooling layer is downsampling the tensor input [47]. The downsampled tensor output is then passed to next set of convolutional layers for feature extraction. In-between the convolutional layers and the dense layers a global average pooling and a dropout layer is implemented. The global average pooling generates a feature map for each corresponding category of the classification in the last convolutional layer

[33]. The following dropout layer is added to regularise the activations to the fully connected layers and hence to improve the generalisation ability and further prevent overfitting [29, 19, 33]. The following 5 fully connected dense layers are also using the ReLu activation function. The dense layers unit size is decreasing with every layer by a factor of 2, starting with 1024 units and ending with 64 units. The last dense ReLu layer passes the tensors to a final fully connected classification layer that uses a softmax activation function with a unit size dependent on the number classes that should be detected in the input data.

The important basic mathematical concepts behind DeepGeoMap's layers in forward propagation, followed by backward propagation is explained step by step in following. Note that the complete mathematical concept behind such a network is too complex for the scope of this work, hence the following will focus on the basic mathematical concepts that are behind the network.

2.1.1 Convolutional Layers

DeepGeoMap uses non causal convolutional layers without padding where the output y is dependent on the future input. These non causal convolutional layers use Cross-correlation by striding (also referred to as sliding) a kernel (filter) across the data. The Filters for each layer are randomly initialized based on a Gaussian distribution. Where the output y is non causal because it is dependent on the future input (x) [18, 49, 28].

The following equation defines the shift of s positions (number of strides) between the input x of length n and the kernel h with a window of length k after each convolution operation for the non-causal convolution [49]:

$$y(n) = \begin{cases} \sum_{i=0}^{k} x(n+i)h(i), & \text{if } n = 0 \\ \sum_{i=0}^{k} x(n+i+(s-1))h(i), & \text{otherwise} \end{cases} \tag{1}$$

The output length of the layer is therefore not equal input length [18, 49, 28]. The length of the output o for stride s is given by equation 2 [49].

$$o = \left\lfloor \frac{(n-k)}{s} \right\rfloor + 1 \tag{2}$$

The convolutional layers of DeepGeoMap use the 'Rectified Linear Unit' (ReLu) activation function. ReLu is used because Rectified Linear Units are computationally undemanding to optimize because they are so similar to linear units [18]. ReLu differ from linear units in that sense that a rectified linear unit outputs zero across half its domain [18]. Hence the derivatives or gradients through a rectified linear unit remain large and consistent whenever the unit is active [18]. Additionally is the second derivative of the rectifying operation 0 almost everywhere, while the derivative of

the rectifying operation is 1 everywhere that the unit is active, making the gradient direction useful for learning and giving an advantage over activation functions that introduce second-order effects [18]. The ReLU input-output relation, where the inner product of the input to a neuron and it's weight-set is denoted by net, and the output function of the neuron is denoted by y, is shown in equation 3 and figure 10 [49].

$$y = \max(0,\ \text{net}\ ) \tag{3}$$

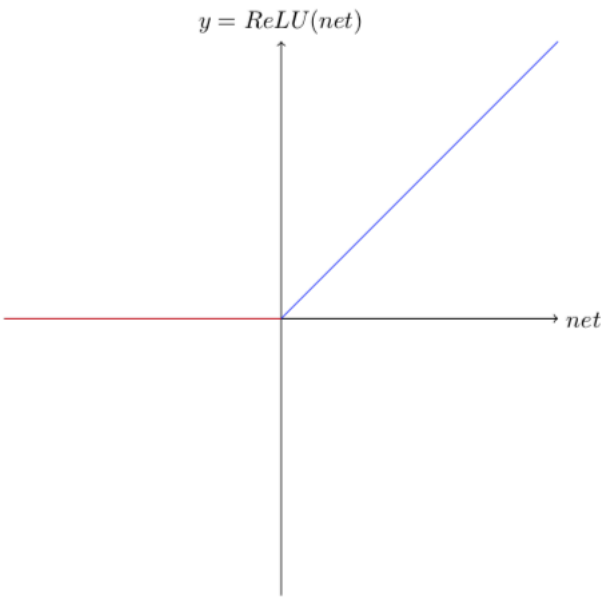

Figure 10: Plot of the ReLU function. Image: by Sakrepatna (2018) [49]

In DeepGeoMap models the input is passed to two consecutive convolutional layers with a kernel size of 3 and filter sizes of 64 and 128 respectively before the output is passed to a Max Pooling Layer.

2.1.2 Max Pooling Layer

The max pooling layer is used to determine the maximum value of the feature maps passed on by the convolutional layers. Hence this reduces the number of feature maps (dimensionality) by maximum pooling over affine feature maps, which also reduces overfitting. The "affine feature maps are the direct results from linear convolution without applying the activation function", Lin (2014) [33]. The "maximization over linear functions makes a piecewise linear approximator which is capable of approximating any convex functions", Lin (2014). Hence, this max pooling makes the network more potent as it is able to separate concepts that lie within convex sets [33]. Consider $Y_{m,k}$ the output of the convolutional layer, x_i the sample number, w_m the kernel size, b the bias, f the activation function, and $Z_{m,L}$ the output of the max pooling layer

then the following functions 4 and 5 apply [34]. The data flow from input over the first two convolutional layers towards the max pooling layer is schematically displayed by Figure 11.

$$Y_{m,k} = f\left(\sum_{i=1}^{n} x_i * w_m + b\right) \tag{4}$$

$$\mathrm{Z}_{m,L} = \max\left(Y_{m,k}\right) \tag{5}$$

In a DeepGeoMap model the dimensionality of the feature maps of the first two convolutional layers is reduced by the factor 5 by the max pooling layer. The max pooling layer then passes the reduced feature maps to another set of 3 consecutive convolutional layers that each grow in filter size with a factor of 2, starting with a filter size of 256, followed by 512 and reaching 1024.

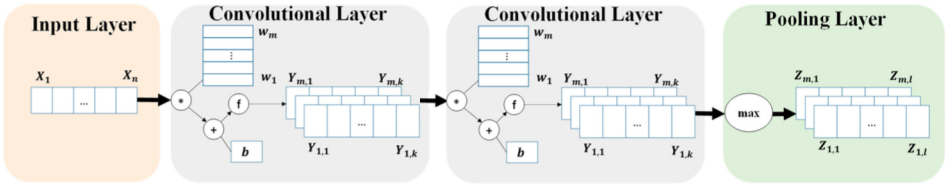

Figure 11: The output feature map of the first two convolutional layers is passed to the max pooling layer. Image modified after Lin (2019) [34].

2.1.3 Global Average Pooling Layer

The idea of a global average pooling layer was first suggested by Lin et al. (2014) as a structural regularizer in their network in network architecture [33]. More traditional convolutional neural networks perform convolution in the lower layers of the network and then passing, the vectorized feature maps of the last convolutional layer directly into fully connected layers followed by a softmax logistic regression layer [33]. DeepGeoMap on the other hand uses global average pooling to add additional classification capability and increase accuracy of the network. The idea is to generate feature maps for the corresponding categories of the classification task in the last convolutional layer, and is inspired by the work of Lin et al. (2014). In a DeepGeoMap model, the global average pooling layer takes the average of each feature map and passes the resulting vector to a dropout layer before it is fed to 5 fully connected layers and a final softmax layer. The advantages of global average pooling over the fully connected layers in regard to categorization of the feature maps, are that they are more native to the convolution structure by enforcing correspondences between feature maps and categories [33]. These feature maps can then be interpreted as categories confidence maps [33]. Additionally there is no parameter optimization in the global average pooling

further reducing overfit of the model [33]. Hence global average pooling is a structural regularizer that explicitly enforces feature maps to be confidence maps of concepts (categories) [33].

2.1.4 Dropout

To further reduce overfitting, and increase the generalization ability of the overall DeepGeoMap network, a dropout layer is added as an additional regularizer. The dropout layer randomly sets half of the activations to the fully connected layers to zero during training. So basically Dropout refers to dropping the output of a neuron which implies zero input to the next layer [33, 49]. In this dropout layer, there can be several neurons that decide to either drop the output or not. This is decided by the dropout rate $P(drop)$ [49]. For each output a random number from 0 to 1 is chosen and if the chosen number is lesser than $P(drop)$ that output will be dropped (figure 12) [49]. In DeepGeoMap the layer has a Dropout rate of $P(drop) = 0.5$, which means every output that chose a number lesser than 0.5 is dropped. In other words this means that randomly half of the output is dropped. Since in DeepGeoMap the Dropout layer follows the global average pooling layer and is followed by the fully connected layers, this means that half of the feature maps passed to the fully connected dense layers is dropped, while inputs not set to 0 are scaled up by 1/(1 - rate) so that the sum over all inputs does not change.

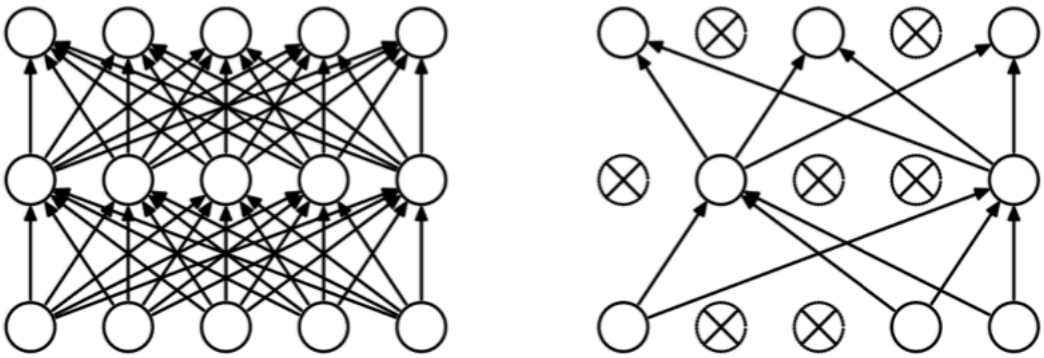

Figure 12: Schematic figure of dropout: Left neurons before dropout, right neurons after dropout applied. Image modified after (Sakrepatna2018), [49]

2.1.5 Fully Connected Dense Layers

Dense layers are densely connected layers that receive information from all the previous layers [20]. This is a principle to ensure maximum information flow between layers in the network, by connecting all dense layers with each other [20]. In order to preserve the feed-forward nature, each dense layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent dense layers [20]. The lth

layer of a dense layer has l inputs, consisting of the feature maps of all preceding layers [20]. This causes $L(L+1)/2$ connections in an L-layer network [20]. One counter-intuitive effect mentioned by Huang et al. (2016) is that a dense connectivity pattern requires fewer parameters than traditional convolutional networks, as there is no need to relearn redundant feature maps. Dense layers also explicitly differentiate between information that is added to the network and information that is preserved [20].

Although dense layers were originally designed for a narrow network (DenseNet) [20], DeepGeoMaps layers are not very narrow. In the first dense layer of DeepGeoMap, 1024 filters are used and then decreasing by factor 2 per layer, adding at first a large set of feature maps to the "collective knowledge" of the network and then focus towards the most important features in the data. This is done in order to extract the maximum amount of information from the preceding layers in order to miss less details of the spectral input, and then use the most important information for classification. In DeepGeoMap also the final classifier, a dense layer with Softmax activation makes a classification decision based on these refined feature maps passed to it. Mathematically this can be expressed modified after Huang et al. (2016) for DeepGeoMap in the following:

Consider a single spectra x_0 that is passed through the network. The network comprises L layers, each of which implements a non-linear transformation $H_l(\,\cdot\,)$, where l indexes the layer. $H_l(\,\cdot\,)$ is a composite function of operations (e.g. rectified linear units (ReLU), Pooling, Convolution, etc.). The output of the lth layer is denoted as x_l.

Then the lth layer receives the feature maps of all preceding dense layers, $x_0, ..., x_{l-1}$, as input:

$$\mathbf{x}_\ell = H_\ell\left([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}]\right) \tag{6}$$

where $[x_0, x_1, ..., x_{l-1}]$ refers to the concatenation of the feature maps produced in layers $0, ..., l-1$ and the multiple inputs of $Hl(\,\cdot\,)$ is displayed as a single tensor.

2.1.6 Gradient Descent Optimization

Gradient descent is a general method in modern neural networks to minimize an objective function by certain parameters [18, 45, 21]. Let this objective function be $J(\theta)$ that is parameterized by the model's parameters $\theta \in Rd$ [45]. Then the function is minimized by updating the parameters in the opposite direction of the gradient of the objective function $\nabla\theta J(\theta)$ w.r.t. to the parameters [45]. Ruder (2016) states that "In other words, we follow the direction of the slope of the surface created by the objective function downhill until we reach a valley."

In the case DeepGeoMap the categorical cross entropy loss function is used to

calculate the losses that are minimized by the gradient decent function. One important hyperparemeter of a decent gradient descent function is the learning rate η, which determines the size of the steps taken to reach a (local) minimum [45]. DeepGeoMap utilzes a learning rate η of 10^{-4}. DeepGeoMap uses a special kind of Stochastic gradient descent function first introduced by by Kingma and Ba in 2014, called the "Adam" optimzer (derived from "Adaptive Moment Estimation") [21]. Stochastic Gradient descent (SDG) performs a parameter update for each training example $x^{(i)}$ and label $y^{(i)}$ where the following function applies:

$$\theta = \theta - \eta \cdot \nabla_\theta J\left(\theta; x^{(i)}; y^{(i)}\right) \tag{7}$$

Adam is a method for efficient stochastic optimization that only requires first-order gradients and hence requires little memory [21]. Adam computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients, therefore the name which is derived from "Adaptive Moment Estimation" [21].

Adam is storing an exponentially decaying average of past squared gradients v_t, as well as an exponentially decaying average of past gradients m_t, (like momentum) [21, 45]:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1)\, g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2)\, g_t^2 \end{aligned} \tag{8}$$

,where g_t to be the gradient of the objective function w.r.t. to the parameter θ at time step t, m_t and v_t are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients respectively [21, 45]. m_t and v_t are initialized as vectors of 0's, they are biased towards zero, especially for the initial time steps, and when the decay rates are small (i.e. $\beta 1$ and $\beta 2$ are close to 1) [21, 45]. To counteract this, the bias-corrected first and second moment estimates are calculated by:

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \end{aligned} \tag{9}$$

The Adam update rule [21] to update the parameters can be then written as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{10}$$

, where default values of 0.9 for $\beta 1$, 0.999 for $\beta 2$, and 10^{-8} for ϵ are set [21].

2.1.7 Loss Function

As mentioned above DeepGeoMap utilizes the Categorical Cross-Entropy loss function which is minimized using the Adam optimizer. First the Cross-Entropy loss is described followed by the Categorical Cross-Entropy loss.

The Cross-Entropy (CE) is defined as:

$$CE = -\sum_{i}^{C} t_i \log(s_i) \tag{11}$$

,where C are the classes, t_i the labels, and s_i the output scores of the network layer, where $f(s_i)$ are the output scores after the activation function is applied. The categorical cross-entropy loss is also sometimes referred to as softmax loss, is basically a softmax activation function plus a cross entropy loss function used for training a neural network for multi-class classification, hence to output a probability over the C classes for each input defined by 12 [17]. In the case of DeepGeoMap output is a probability over C classes for each pixel spectrum.

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \quad CE = -\sum_{i}^{C} t_i \log(f(s)_i) \tag{12}$$

In multi-class classification the labels are one-hot, which means that only the positive class C_p keeps its term in the loss. Only one element of the target vector t is not zero $t_i = t_p$. Hence the elements of the summation which are zero due to target labels are discarded and function 13 applies, where S_p is the score for the positive class [17].

$$CE = -\log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right) \tag{13}$$

How Adam back-propagates through the net is shown above. Hence the gradient of cross cntropy (CE) loss in respect to each class score in s is required by the optimizer. The loss terms from the negative classes are zero. However, since the softmax activation of the positive class also depends on the negative class scores the loss gradient in respect of those negative classes is not cancelled [17].

The gradient for all classes C is equal, with the exception of the ground truth class C_p do to the score of $C_p(s_p)$ in the nominator [17]. Out of this follows that the derivative in respect to the positive class is:

$$\frac{\partial}{\partial s_p}\left(-\log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)\right) = \left(\frac{e^{s_p}}{\sum_j^C e^{s_j}} - 1\right) \tag{14}$$

,while the derivative in respect to the negative classes C, with score s_p is:

$$\frac{\partial}{\partial s_n}\left(-\log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)\right) = \left(\frac{e^{s_n}}{\sum_i^C e^{s_j}}\right) \tag{15}$$

2.2 Samples and Data Background

For the development of the proposed deep convolutional neural network architecture only geochemically verified (Brazilian iron ore Samples, and Apliki samples) or well known ground truthed hyperspectral images (the Indian Pines Dataset) were used. The Brazilian iron ore and Apliki Samples were measured at the optical spectroscopy Laboratory at the GFZ Potsdam and in the field in the Republic of Cyprus by Friederike Koerting for her dissertation using the Neo Norsk Electric HySpex VNIR-1600 and SWIR 320-m-e Camera System (see section 2.2.1 and 2.2.3) [24]. The Brazilian iron ore and the Apliki sample data sets are both part of Friederike Koertings dissertation ([24]) to compare and in-depth analyse different geological classification algorithms and methods. The data sets are very beneficial in regard to being very well geologically analysed, geochemically validated, and hyper-spectrally benchmarked datasets. This allows a comparison of the classification accuracies of DeepGeoMap and some of the algorithms used by Friederie Koerting in her dissertation. The Indian Pines dataset is a well known dataset in the machine learning community used in various publications to benchmark and compare image classification capability of algorithms and neural networks, it was recorded by NASA using the Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) (see section 2.2.4) [6, 41]. In the following sections the background of each of the sample sets and the hyperspectral data used is presented.

2.2.1 Brazilian Iron Ore Samples

The hyperspectral image of 15 Brazilian iron ore samples that was analysed with DeepGeoMap, was provided and recorded in the optical spectroscopy Laboratory of the GFZ by Frederike Koerting in the context of her research and dissertation (see [24]). The majority of the geological background research is provided in the dissertation [24]. The hyperspectral laboratory scan includes 15 iron ore samples from two mine sites in Brazil[24]. It is stated that the samples and their respective geochemical analysis were provided under a selective non-disclosure agreement by clients from the company rad. Data Spectral analytics (www.raddata.io, 2020), only information important to classification benchmarking is provided [24]. The mining companies supplying the samples from their sites are not to be disclosed, neither are the exact locations of sampling and therefore the collection sites are referred to as Mine Site 1 and Mine Site 2 [24]. The sample numbers, hyperspectral images and geochemistry are shown in figure 13 and table 2 [24].

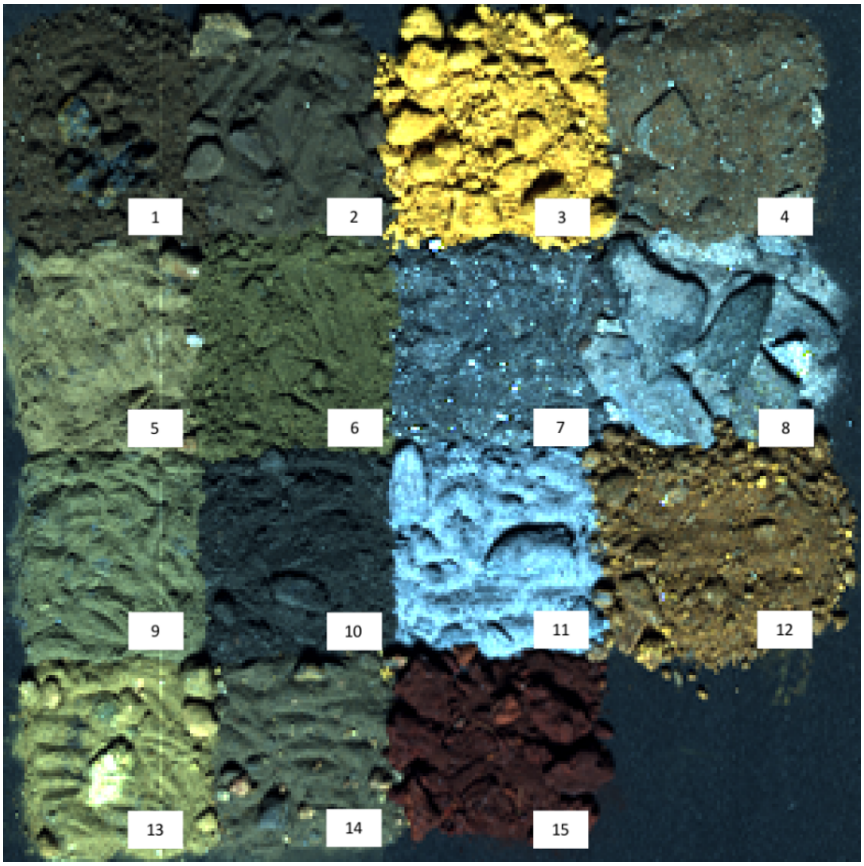

Figure 13: Brazilian iron ore samples in RGB color (R: 640nm - band 63, G: 549nm - band 38, B: 458nm - band 13), recorded by the HySpex VNIR camera. The labels show the sample numbers, the according geochemistry is shown in table 2 (after Koerting, 2021) [24].

Geological Background: The 15 iron ore samples from this dataset originate from the Quadrilátero Ferrífero (QF) in the state of Minas Gerais, a southeastern region of Brazil from two Mine sites [24]. The QF spans an area of about 7000 km^2 and hosts one of the largest concentrations of lateritic iron-ore deposits in the world [8]. The QF is structurally controlled and square-shaped due to the dome and basin structures of the Minas Supergroup (see figure 14) [8, 10].

The supergene iron ores of Minas Gerais and QF can be divided into three main categories: enriched itabirite, intermediate grade ore and Canga [8, 12]. The majority of the iron ore found in the QF are hosted in the oxidized, metamorphosed and heterogeneously deformed Banded Iron Formations (BIFs) of the Cauê Formation, the so-called itabirites [8]. Carlos et al. (2014) describe the BIFs as "altered sedimentary deposits with laminated rocks formed by alternating layers of silica and hematite–magnetite, as well as carbonates and iron silicates" [8]. Itabirite is described by Dorr (1964) as a "laminated, metamorphosed, oxide-facies iron formation in which the iron is present as hematite, magnetite or martite" [12, 24]. In itabirite, the original jasper or chert bands have been recrystallized into granular quartz [12]. Locally, dolomite and amphiboles can be substitutes for quartz and when substituted, magnetite and its

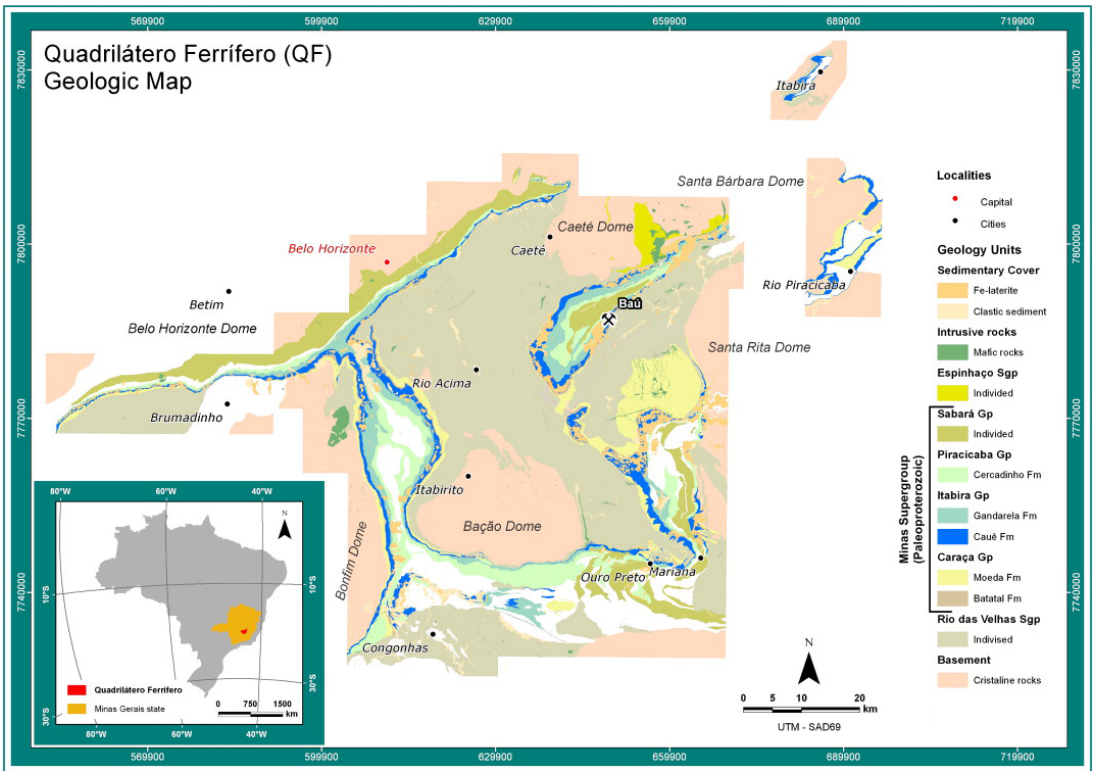

Figure 14: "Regional geological map of the Quadrilátero Ferrífero area located at Minas Gerais state, Brazil", (modified after Carlos et al. (2014)) [8]

oxidation products are present [12]. Fresh Itabirite is a dense, brittle and hard rock, that is resistant to mechanical erosion [12]. All of the supergene iron ores found in the QF are a result from the weathering of itabirite [12]. They degrade based on the same geological processes, where the only difference is the varying intensity of weathering [12]. Hence the iron ores are intergradational and cannot always be clearly put in only one category as they represent different stages or intermediate stages of weathering [12]. Enriched itabirite is defined as a disaggregated rock which is residually enriched in iron and leached of quartz and other components by supergene fluids [12]. The intermediate grade ore found in the QF is a material with an upper limit of iron content of about 65.5% and a lower limit of 57% iron and <7% SiO2, which is derived from strong residual and secondary enrichment of itabirite [8, 12]. Canga has a high iron, a low silicate content, is resistant to erosion and chemical weathering and composed of varying detrital materials derived from limonite cemented high-grade hematite ore or itabirite [12]. It forms as consolidated, extensive blanket deposits at or near erosion surfaces [8]. The definition and composition of itabirite and its weathering products by Dorr (1964) is shown in Table 1 [12]. With the definitions by Dorr (1964) and the Mine site information the 15 iron ore samples were categorised by Koerting (2021) shown in Table 2.

Table 1: Bulk rock composition of itabirite and its weathering products Dorr (1964), after Koerting (2021) ([12, 24])

Element or Oxide	Unleached Itabirite [$wt.\%$]	Enriched Itabirite [$wt.\%$]	Intermediate grade ore [$wt.\%$]	Canga [$wt.\%$]
Fe	37.9	48.7	63.3	62.2
Fe_2O3	54.1	69.6	90.4	88.8
SiO_2	44.7	25.4	2.35	1.7
Al_2O_3	0.5	1.3	2.6	2.8
P	0.05	0.06	0.08	0.1
H_2O plus	0.3	/	4.6	5.4

Table 2: Geochemical composition and categorization of the 15 iron ore samples from Koerting (2021) after Dorr (1964) ([12, 24])

ID	Fe	SiO_2	Al_2O_3	P	Categories after Dorr, 1964	Categories based on Mine Site 2	Mine Site No.
1	55,05	16,81	0,61	0,14	Itabirite	Medium value ore	2
2	49,72	18,51	1,05	0,11	Itabirite	Medium value ore	2
3	24,40	21,69	22,14	0,60	Itabirite, high clay contamination	Low value ore	1
4	46,39	30,33	1,41	0,04	Itabirite	Medium value ore	1
5	38,06	41,16	0,43	0,06	Itabirite	Low value ore	2
6	44,47	31,59	1,97	0,10	Itabirite	Medium value ore	2
7	58,22	10,14	3,94	0,09	Enriched Itabirite	High value ore	1
8	35,37	49,02	0,28	0,02	Itabirite	Low value ore	1
9	44,46	32,99	0,53	0,08	Itabirite	Medium value ore	2
10	68,49	0,53	0,45	0,05	Canga	High value ore	2
11	30,58	56,5	<0,05	<0,005	Itabirite Low	value ore	1
12	47,92	13,29	9,26	0,21	Itabirite	Medium value ore	1
13	37,17	41,64	1,29	0,18	Itabirite	Low value ore	2
14	38,23	37,45	1,54	0,05	Itabirite	Low value ore, high Mn contamination	2
15	64,93	1,32	1,48	0,11	Intermediate grade ore	High value ore	1

Image acquisition and processing: Mine Site 1 delivered 8 samples (sample IDs: 3, 4, 7, 8, 11, 12 and 15), while Mine Site 2 delivered 7 samples (sample IDs: 1, 2, 5, 6, 9, 10, 13 and 14) [24]. For the measurements the 15 samples were left to dry, and then arranged in 4x4cm squares on a black rubber lining [24]. The samples were then scanned without any further preparation with the Hyspex Cameras [24]. For laboratory measurements the HySpex VNIR-1600 and SWIR-320m-e (pushbroom line-scanning cameras) are mounted parallel in a frame (figure 16 . Images are acquired

in the visible to near infrared (VNIR, 414 – 1000nm) and the shortwave infrared (SWIR, 1000 – 2498nm). They record an array-line of 1600 pixel (VNIR) and 320 pixel (SWIR). Every pixel contains a spectrum with a total spectral sampling number of 408 bands in total. In pushbroom cameras the field of view (FOV) captures the spectral dimension (λ) for each pixel of a line of fixed pixels (y-dimension), while the x-dimension is acquired by subsequent lines following the movement perpendicular to the y-dimension [24, 44]. For image acquisition, the sensor or the objects in the FOV can move perpendicular to the y-dimension (FOV) [24, 44]. In the laboratory the movement of the samples is accomplished by the movement of a special translation stage as seen in figure 16 [24, 44]. The stage moves the samples through the FOV in accordance to the integration times of the HySpex Cameras while a light source illuminates them in a certain angle [44]. A white reference panel, is placed in line with the samples, and its reflectance level is chosen according to the albedo of the samples [44, 24]. Both samples included in this thesis, The Brazilian iron ore samples and the Apliki mine samples required a 50% reflectance white reference panel [24]. The light source was set to an angle of 45°between incident light and the vertical plane [24, 44]. The software specifically designed for the HySpex instruments 'HySpex ground' is used to perform the measurements and the software 'HySpex rad' is used to perform the radiometric calibration of the acquired images [24]. The hyperspectral image cubes (VNIR and SWIR) are then co-registered, resized and stacked to a continuous image cube [24]. For reflectance retrieval, the image pixels that show the white standard were averaged to a one-line reference spectrum, and then the reflectance was calculated by dividing every image line spectrum by its reference spectrum from the reflecting white reference panel [24, 44]. After the reflectance retrieval, the data was cropped to the spatial extend of the samples [24].

The spectral signatures of the 15 samples are shown in figure 15.

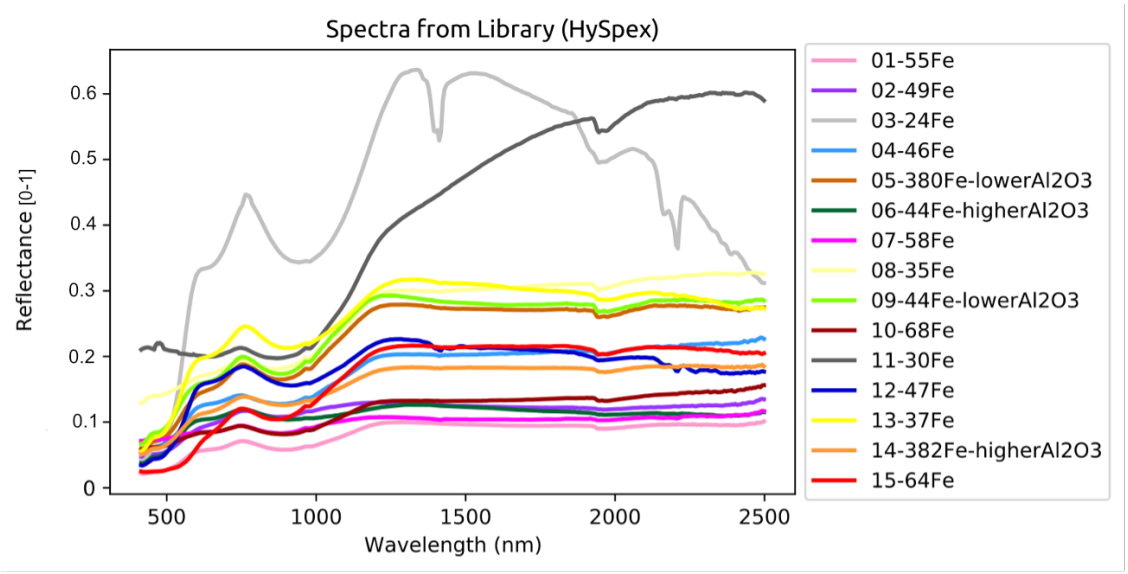

Figure 15: The spectral reflectance signatures of the 15 samples derived from a 5x5 pixel averaging. [24] Image: Koerting (2021)

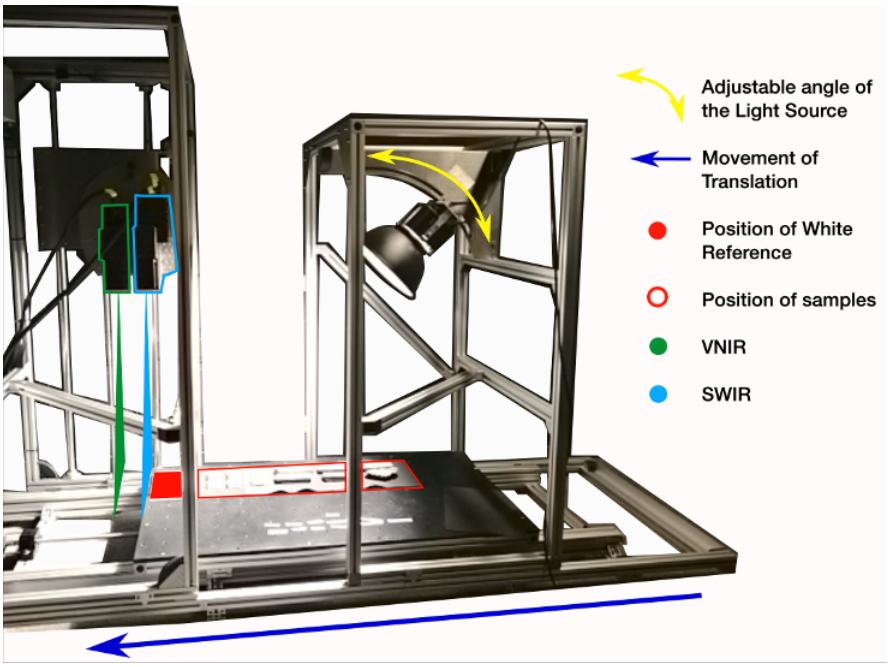

Figure 16: The HySpex translation stage in the optical laboratory of the GFZ. The HySpex cameras are mounted in place, while the translation stage moves the samples through the FOV with a speed in accordance to the integration time of the two cameras. [44, 24, 26] Image: Koerting (2019)

2.2.2 Apliki Field Scans

It has to be stated that: All the work in the Republic of Cyprus was conducted under the Permit to conduct a Geological Survey, Ref. No. 02.13.005.002.005.022 from the 19th of March 2018, granted by the Geological Survey Department Cyprus, Ministry of Agriculture, Rural Development and Environment (GSD) and the Director Dr. Costas Constantinou. The permit terminated in 18th of September 2018. An agreement for

a Memorandum of Understanding and Framework (MoU) for cooperation in the area of geo-science between the GSD and the Helmholtz Centre Potsdam (GFZ) German Research Centre for Geosciences was reached in March 2019 including the objective of "hyperspectral mapping of secondary minerals in the field of existing drill cores e.g. in abandoned mines or for the purpose of environmental monitoring". This work is linked to both, the field sampling permit from March 2018 and the MoU agreement reached in March 2019.

The Apliki mine in the Republic of Cyprus is a copper-gold-pyrite mine that was surveyed, hyperspecrally scanned, and sampled by the Geological Survey Department of the Republic of Cyprus (GSD) and the German Research Centre for Geosciences (GFZ)) in March 2018 [24, 26, 25].The Apliki VMS outcrop was scanned with two hyperspectral cameras (HySpex VNIR-1600 and SWIR-320-m-e) and then sampled [24]. For more in-depth descriptions of the data and acquisition please consider Koerting (2021) and Koerting (2019) found in the References [24, 26, 25].

Geological Background: The Apliki mine is a structurally controlled Cyprus-type massive sulphide deposit confined by two axis parallel (N-S) normal faults, and is located within the Lower Pillow Lava stratigraphy of Cyprus (figure 18)[24]. The Apliki deposit is a typical example of a Cyprus-type volcanogenic massive sulfides (VMS) Cu-pyrite ore deposit that was abandoned in the mid-1970s [3, 24]. It is described as a massive sulfide lens that is superimposed by a cupriferous stockwork of quartz, jasper and sulfide veins within the basaltic host Lower Pillow Lavas [4, 35, 24]. The estimates for the grade of the copper ore range between 0.01 to 3.5wt.% Cu, Gold grade does not exceed 0.1 g/t, and sulfur between 0.1 to 16 wt.% [4, 2, 24].

The fault-bound zone of mineralization is circa 100 m wide and consists of silicified, chloritized, brecciated Lower Pillow Lava with some scattered pyrite absent of larger sulfide mineralizations [24]. The eastern and western parts of the cut dominantly show unmineralized hyaloclastic Pillow Lavas, while lava east of the fault is increasingly chloritized and quartz is accompanied by analcime as vesicle-filling material [24]. The Northern wall and open cut area represents the contact with the ore zone [4, 24]. The open cut area is around 150 m high and 150 m wide at the time of visit, while the pit bottom is around 200 m a.s.l [4, 24]. Surface water flow caused sulfate evaporates, possibly chalcanthite, to cover parts of the cut terraces, while satin spar gypsum veining occurs at the northern part of the open cut [24]. The cupriferous massive sulfide ore has been mined in Apliki making the mine "a typical example of an oxidation zone overlying a stockwork type sulfide mineralization in chlorite-bearing and silicified lavas" [24, 4]. This massive sulfide was originally on top of the stockwork zone, covered by Upper Pillow Lavas and tuffaceous sediments, while some of the ore zone continues northwards as low-grade disseminated mineralization [24]. The typical stockwork type

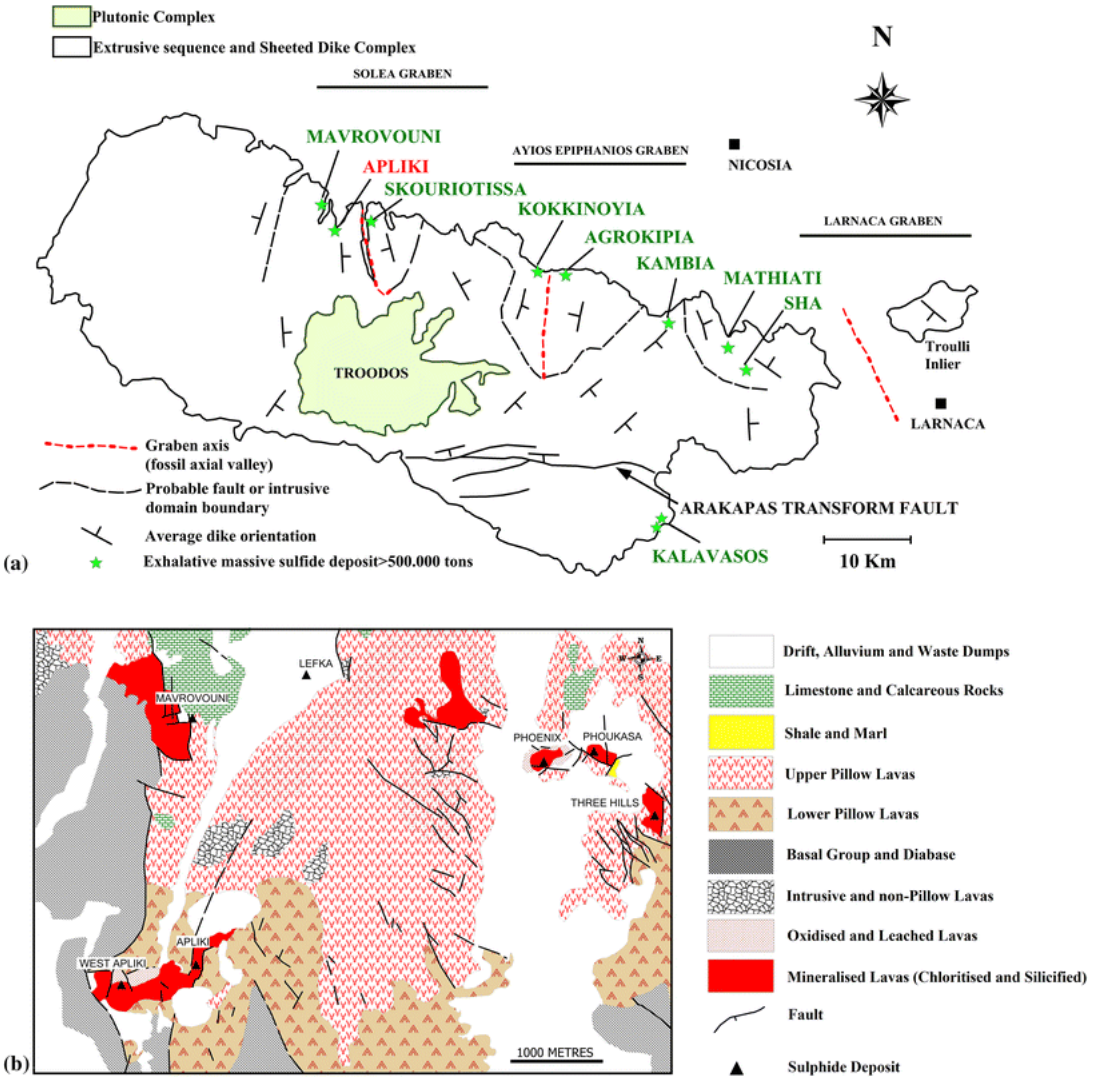

Figure 17: (a): Generalized geological map of the Republic Cyprus showing location of the main massive sulphide deposits and the relative extent of the plutonic and the sheeted dike complex and extrusive sequence. The average orientation of the diabas dikes are indicated [**Vagra1985**]. The Apliki mine is marked red. (b) Geological map of the Skouriotissa–Mavrovouni region [2]. Image after: [4]

mineralization shows cavity and vein fillings of pyrite and subordinate chalcopyrite in the southern part of the open pit while altered lavas feature jasper [24]. Koerting (2021) describes five facies within the mineralized zone after Antivachis (2015) as following: "1) a stockwork zone, 2) veins of amorphous silica in the NNW, 3) massive veins of Gypsum in the NNW, 4) the "red zone", an oxidized vein of mineralization and 5) the oxidation zone at the top" (figure 18) [24, 4].

The hyperspectral scans shows 5 ½ levels of in the Northern part of Apliki VMS, situated in the fault bound zone of chloritized-silicified brecciated Pillow Lava [24]. The so called stockwork zone or "mineralization zone" by Antivachis (2015) in Figure 18 takes up the largest area captured by the hyperspectral sensors [24]. This stockwork zone is dominated by chlorite, clay minerals, albite, iron and titanium oxides and

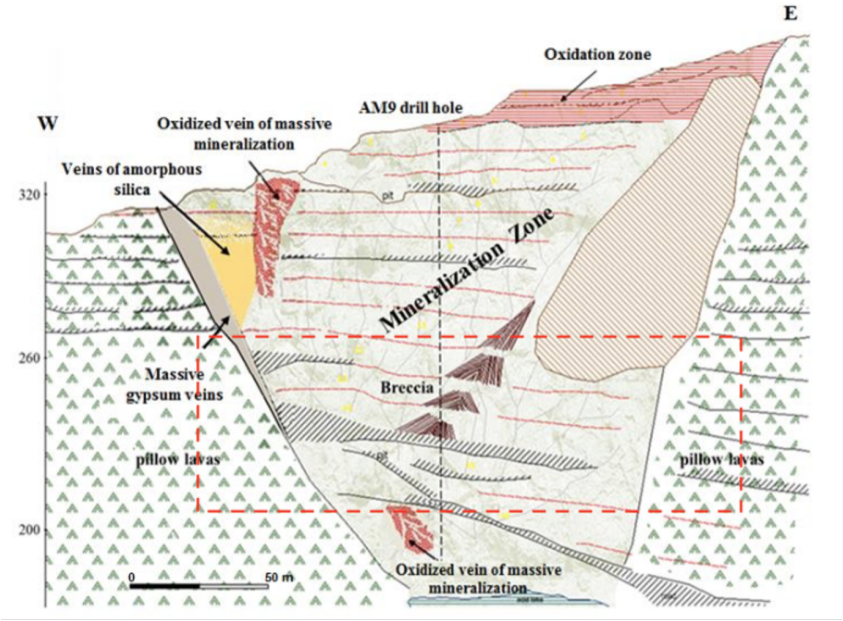

Figure 18: The schematic geological section of the Northern part of Apliki mine from Antivachis (2015). The red, dashed window shows the part covered by the hyperspectral mine face scan (see Koerting (2021)). For the hyperspectral mine face scan see section 2.2.2. Image: Koerting (2021) after Antivachis (2015).

quartz [24]. Often fractured pyrite shows fills of chlorite [24]. In the chloritized ground mass volcanic glass is replaced by clay minerals, while Magmatic plagioclase was albitized and formed pure albite [24]. Calcite often acts as filling material within the groundmass and partially replaces Plagioclase and pyroxene, while anatase replaces Iron-titanium oxides and ilmenite [24]. Koerting (2021) spatially classified after Antivachis (2015) the prevailing alteration (chloritization) in the mineralized zone into three subzones: "1) Smectitic: montmorillonite, illite and limited chlorite; 2) Chloritic-smectitic: stronger chlorite content as replacement of pyroxenes; 3) Chloritic: magmatic minerals are absent, chlorite dominates (Antivachis, 2015)" [24]. The brecciated lavas are primarily made of quartz, goethite and hematite with minor amounts of clays and carbonates and jarosite, additionally Jasper is present widespread [4, 24]. The sulfides are dominated by pyrite and minor sphalerite and chalcopyrite which often shows inclusions of sphalerite [24]. Chalcopyrite is also sometimes replaced by covellite and chalcocite due to supergene processes [24]. The Predominant ore minerals present are pyrite, marcasite and chalcopyrite, with additons of bornite, sphalerite, galena and barite [24].

Image aquisition and processing: Two hyperspectral cameras (HySpex VNIR-1600 and SWIR-320-m-e) were used in simultaneous recoding mode on a rotational stage to conduct the measurements [24]. The HySpex cameras were placed approximately parallel to the outcrop face at a distance of 100 m SEE to the mine face [24].

Several white refence targets were place in the FOV of the cameras in order to retrieve hyperspectral reflectance images.

The pre-processing for the hyperspectral reflectance images by Koerting (2021) is described in the following. The raw Digital Number (DN) sensor output data was radiometrically scaled to radiance using manufacturer predefined sensor characteristic radiometric calibration coefficients [24]. The mean radiance spectrum of each reflectance target was calculated and normalized according to their relative reflectance level [24]. An irradiance model was calculated for each target and is applied to every HSI pixel to retrieve the reflectance value for each band [24]. Overall eleven data cubes were averaged and combined into one reflectance dataset that was further processed and downsampled [24]. The original VNIR-SWIR scan has a size of 3013 x 320 pixel, including parts of the open pit in the Southeast [24]. The scan was clipped to the extent of the relevant mine face area to 1600 x 320 pixel [24]. It is to note that Koerting (2021) used a reflectance retrieval routine by Christian Rogaß (Boesche et al., 2015) that includes a smoothing with a Gaussian filter with a sigma of 2, while in this Master-Thesis the same routine is used without the gaussian smoothing. Further the data was clipped to the wavelength range of 414 – 2390 nm, excluding the last 18 bands due to prevalent noise [24]. The atmospheric bands, were clipped between bands 209 – 239 and bands 284- 327 [24]. Finally a shadow and Normalized Difference Vegetation Index (NDVI) masking was performed based on the Indexdatabase (indexdatabase.de/, 2020) [24]. The resulting image has 390 bands and its RGB representation is shown in figure 19.

Figure 19: "NDVI & shadow masked reflectance field scan. Vegetation and areas of shadow (high SNR) are masked out and shown in black. (RGB composition: R: band 63, G: band 38, B: band 13.)", Koerting (2021) [24].

2.2.3 Apliki Lab Scans

The Laboratory dataset includes 36 samples collected in in March 2018 in the copper-gold-pyrite mine Apliki in the Republic of Cyprus by the Geological Survey Department of the Republic of Cyprus (GSD) and the German Research Centre for Geosciences (GFZ)) [24, 26]. The samples were geochemically analyzed and the data and the according spectra were published in 2019 by Koerting et al. [24, 26]. A hyperspectral image mosaic containing the 36 samples from 14 different sampling locations on the

mine face was created by combining (stichting) a total of 9 hyperspectral laboratory scans/images together [24]. The Original dataset/mosaic contained 44 samples but only 36 from the 44 original samples were geochemically analysed (and hence verified) and published by Koerting et al. in 2019 [24, 26].

Sampling locations: Koerting (2021) identified spectrally homogenous regions for sampling, using a Principal Components Analysis (PCA) from the SWIR radiance data [24]. The RGB false color of the PCA band 3-5-6 is shown in Figure 20 [24]. "Field sampling was conducted in the areas indicated by red stars in the PCA. Thirty-seven samples were collected on–site from fourteen different sampling areas" (see Figure21) [24]. Several samples have been collected from each sample area and were analyzed spectrally and geochemically as described in the laboratory section of the work by Koerting (2021) and Koerting et al. (2019) [24, 26].

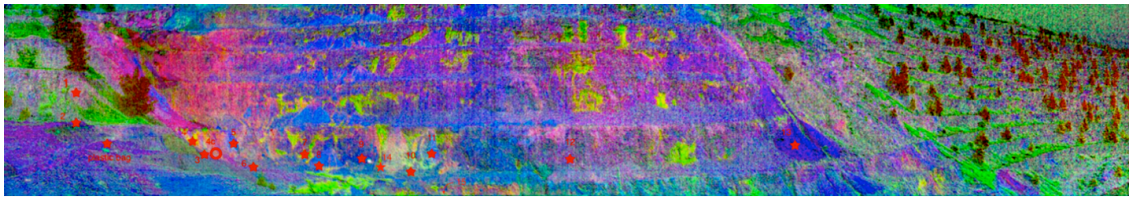

Figure 20: "PCA based on SWIR radiance data marking spectrally homogeneous regions with red stars for possible field sampling.)", Koerting (2021) [24].

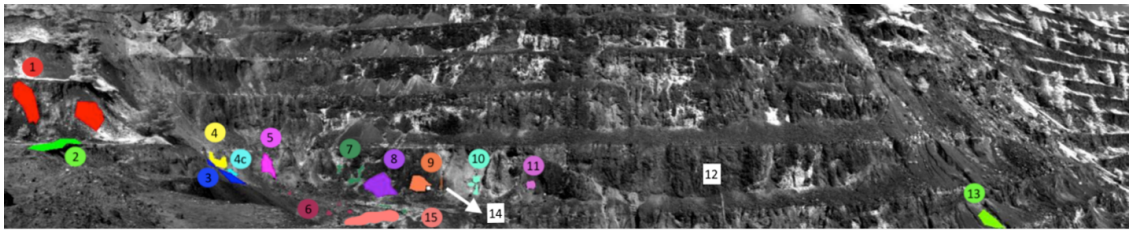

Figure 21: "Sampling locations identified from SWIR radiance PCA analysis and marked as ROIs in the SWIR grey-scale image Sample area 1-11, 13 and 15 were included in the spectral characterization of the mine face (marked in their ROI color), area 12 & 14 were excluded (labeled in white)", Koerting (2021) [24].

Image acquisition and processing: In order for the laboratory scans to represent most exactly what was found in the field, the samples were left in their natural state: "partly dusty, showing heterogeneity in the small scale e.g. differently colored soil grains from the same sample location, inhomogeneous tarnish across individual samples and in a variety of states (e.g. soil, crystallized evaporitic material, fresh rock surface, tarnished rock, oxidation zones)" [24]. The samples were placed in black, plastic containers that were previously sprayed with matt black spray-paint to avoid backscattering of plastic features from the box surface onto the samples [24]. The

samples made of whole rocks, were placed in the same angles/directions at the sensor as found in the field and hence as present as in the according field scans (see Section 2.2.2 [24]. The same hyperspectral cameras as in the field measurements were used (HySpex VNIR-1600 and SWIR-320-m-e) but instead of a rotational mount the cameras are mounted in a special translation stage build for laboratory measurements figure 16 [24]. The samples where scanned with the same method as described in section 2.2.1 with the exception of using different integration times [24]. A hyperspectral image mosaic containing the 36 samples from 14 different sampling locations on the mine face was created by combining (stichting) a total of 9 hyperspectral laboratory scans/images together, 8 bad noisy bands were cut resulting in 400 bands (figure 22) [24].

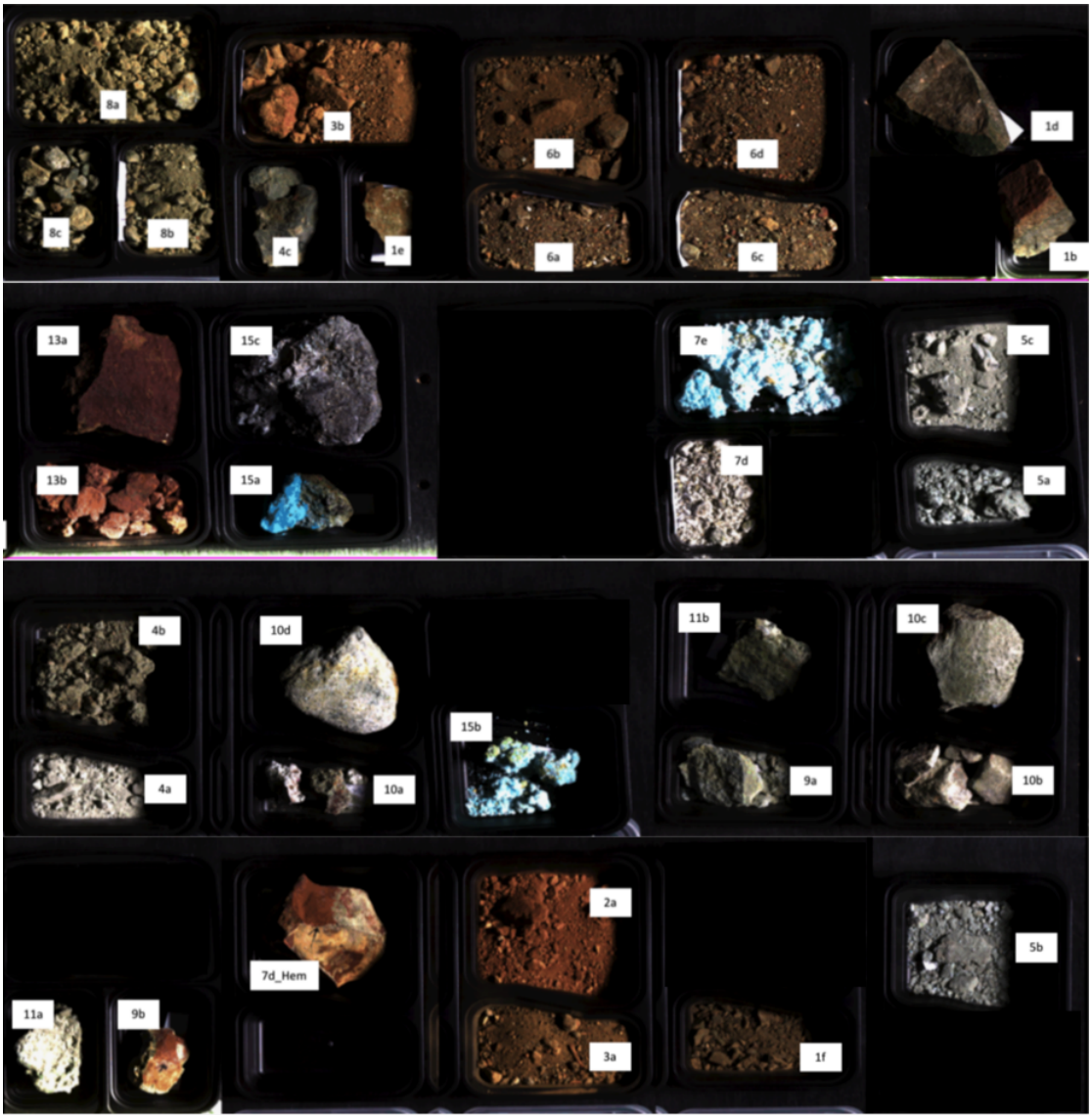

Figure 22: Apliki mine samples hyperspectral mosaic of 9 images stiched together (R: 640nm – band 63, G: 549nm – band 38, B: 458nm – band 13) the sample numbers are shown in the white boxes, Image: Koerting (2021) [24].

Sample clustering: Koerting (2021) used geochemical and spectral analysis to cluster the sample data [24]. First the VNIR-SWIR reflectance data of the Apliki mine sample scans were analysed [24]. 7 clusters were determined to describe the geochemical variance of the data in the optimal way (see figure 23) [24]. For this the full geochemistry of 36 samples was used for the clustering approach [24]. The spectral reflectance profiles for the 7 cluster are shown in figure 24. For a more detailed description please consider Koerting (2021) [24].

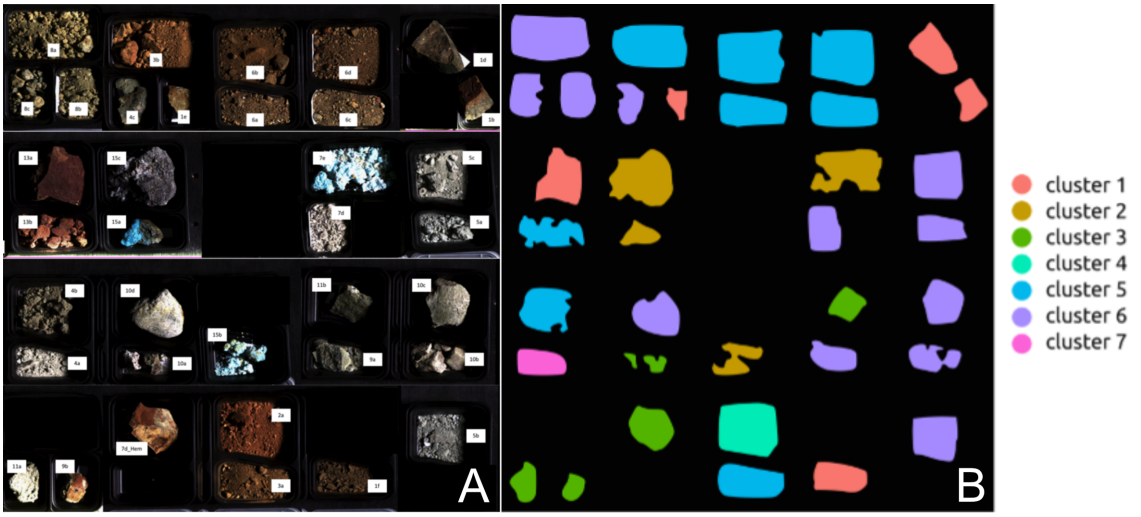

Figure 23: A; Apliki mine samples hyperspectral mosaic of 9 images stiched together (R: 640nm – band 63, G: 549nm – band 38, B: 458nm – band 13) and B; The samples coloured to display cluster affiliation, Images: Koerting (2021) [24].

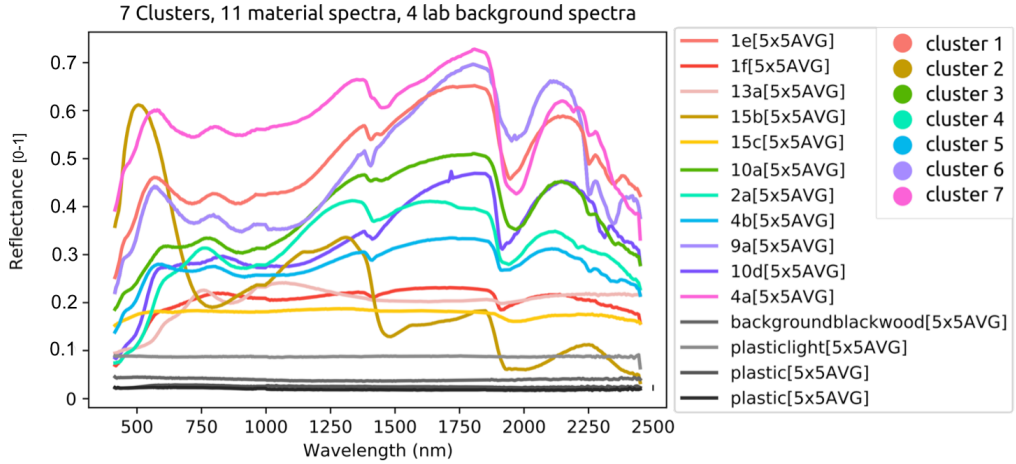

Figure 24: The spectral reflectance signatures of the 7 sample clusters derived from a 5x5 pixel averaging of each indicated sample [24]. Image modified: Koerting (2019) [26]

Table 3: "Apliki mine sample clusters, associated sample IDs and cluster mineralogy.", Koerting (2021) [24]

Cluster	Samples	Cluster Mineralogy
01	1b, 1d, 1e, 1f, 13a	Quartz, **Plagioclase Feldspar** (Andesine, Anorthite), **Pyroxene** (Diopside), **Smectite-group:** Montmorillonite, **Fe-Oxide:** Magnetite, **Sulfide:** (Pyrite (Fe, one sample only)); Dominated by: Plagioclase and Montmorillonite.
02	15c, 15a, 15b, 7e	Quartz, **Fe-Oxide:** Goethite, **Sulfides** (Cu, Fe, CuFe): Covellite, Pyrite, Chalcopyrite, **Sulfates** (Cu, Fe, Mn-Al, Mg): Chalcanthite, Ferrohexhydrite, Apjohnite, Rozenite, Pentahydrate (cuprian); Dominated by: Quartz, Sulfates and Sulfides, Fe-Hydrate (7e)
03	11a, 11b, 10a, 7d_hem, 9b	Quartz; Fe-Oxide: Goethite; **Sulfides** (Cu): Pyrite; **Sulfates (Cu, Zn-Fe):** Gypsum, Bassanite, Sphalerite; **Chlorite group:** Clinochlore; Dominated by: Quartz (+ Chlorite-group (sample 11a, 11b))
04	2a	**Fe-Oxide:** Goethite; **Sulfate (K-Fe):** Jarosite-Natrojarosite; Quartz; **Plagioclase Feldspar** (Andesine); **Chlorite Group**: Clinochlore; Dominated by: Sulfates
05	13b, 3a, 3b, 4b, 6b, 6c, 6a, 6d	Quartz; **Plagioclase** (Andesine, Anorthite); Analcime; **Pyroxene** (Diopside); **Smectite-group:** Montmorillonite; **Fe-Oxide:** Goethite, Magnetite; **Sulfate (K-Fe, Ca):** Jarosite, Gypsum; **Chlorite-group:** Clinochlore; **Sulfide (Fe):** Pyrite; Dominated by: Clays, Smectite-chlorite group
06	4c, 5a, 5b, 5c, 8a, 8b, 8c, 9a, 10b, 10c, 10d, 7d	Chlorite-group: Clinochlore; Smectite-group: Montmorillonite; Sulfate (Ca, Mg): Gypsum, Hexahydrite; Quartz; Sulfide: Pyrite; Fe-Oxide: Goethite; Ajoite (minor copper ore, silicate hydroxide); Dominated by: Chlorite-group
07	4a	Sulfate (Ca, Fe): Gypsum, Rozenite; Quartz, Chlorite-group: Clinochlore; Dominated by: Gypsum

2.2.4 Indian Pines Dataset

The indian pines dataset can generally be seen as a benchmark data set for hyperspectral image classification algorithms and methods including machine learning [41]. This scene was gathered by the National Aeronautics and Space Administration (NASA) with the Airborne Visible/Infrared Imaging Spectrometer AVIRIS sensor over the Indian Pines test site in North-western Indiana (USA) as a subset of a larger scene [30, 6, 41]. It consists of 145×145 pixels with a spatial resolution of 20 m and 224 spectral reflectance bands in the wavelength range 400-2500 nano meters [30, 6, 41]. The Indian Pines scene contains around two-thirds agriculture, and one-third forest or other natural perennial vegetation and includes two major dual lane highways, a rail line, as well as some low density housing, other built structures, and smaller roads [30, 6, 41]. The scene was acquired in June and some of the crops such as corn, soybeans, are in early stages of growth with less than 5% coverage [30, 6]. The ground truth included in the data is divided into 16 labeled classes (figure 25).

The Indian Pines data is made available through Pursue's univeristy MultiSpec website [6].

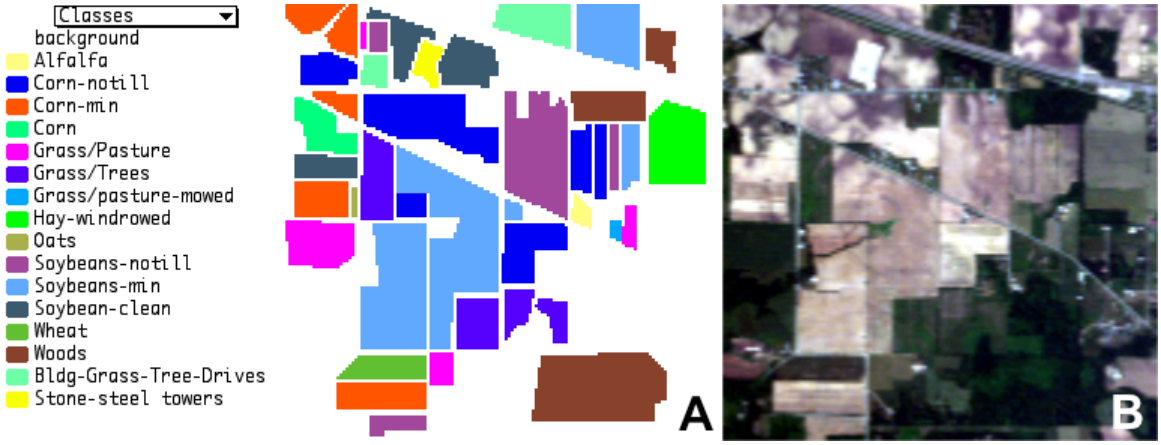

Figure 25: A; The Indian pines ground truth with classes in the original color-scheme, and B; an approximate true color image of the scene (Red: band 27 (646.7 nm), Green band 17 (547.6 nm), Blue band 7 (449.1 nm)) [6]

2.3 DeepGeoMap: Data pre-processing/preparation, and computation

The image pre-processing of the 2 different data sets used to train DeepGeoMap is described in the following. Partly this was done using the image processing and analysis software ENVI 5.5.3 from L3Harris Geospatial and partly with the programming language Python 3.7 [16]. Note that the Apliki field image data (mine face scan) was classified using a model that was trained with the Apliki laboratory data (see Section 3).

2.3.1 Brazilian Iron Ore:

The hyperspectral reflectance images of the Brazilian iron ore samples were first loaded into ENVI, where for each of the 15 samples a ROI (Region of Interest) was created (see figure 26. From here on the 15 different sample types will be interchangeably treated and (sometimes) referred to as classes (e.g. sample 1 = class 1, sample 2 = class 2 etc.). One additional ROI was added covering the area of the black cellular rubber material that the samples were placed on, creating a 16th class. This was done so that the DeepGeoMap model can learn the spectral features of the background in order to make a mask for classification obsolete. The original image has the spatial dimensions of 280x280 pixels. The ROIs are square shaped (51x51 pixels) and roughly covering the middle of each sample without being too close to any neighbouring sample. This is one important aspect of DeepGeoMaps data pre-processing as this ensures that the class labeled training data is only created from pixels that cover a validated sample (in this case also geochemically validated) and no bordering/mixed pixels are used for training the models. The ROIs are also supposed to include shadowed areas of the sample material as the DeepGeoMap model will then learn to identify the class

of a pixel that is illuminated not ideally (which is often the case for mine face data acquired in the field). The ROIs of each class are then used to extract a spatial subset of the extent of the ROI for each class, resulting in 16 individual hyperspectral images with dimensions of 51x51 pixel containing the full spectral range of 408 bands. The resulting 16 images where then stitched to a mosaic containing all of the images while retaining the same order as in the original image. This mosaic of the 16 sample classes is shown in figure 27. The spectral data of this mosaic is then used as training data input for the DeepGeoMap model. Since DeepGeoMap is a 1D neural network, and only the spectra are relevant, the mosaic image data needs to be flattened. This results in 51x51 = 2601 training pixels per class and a total of 16x2601=41616 training pixels with 408 spectral bands for the dataset. The labelling of the data is done either by creating a label array of the size of the mosaic with the according summed length of each class, where for each class 2601 labels are set, or by creating an image label mask (which is later turned into an array anyway). The image label mask is shown in figure 28, it is simply an image containing grey scale values for the samples/classes 0-15 (including 0, as Python starts counting with 0). This label image feature was added as it seemed more intuitive and user friendly and allows a better overview of how the classes are distributed and can easily be compared to the mosaic image. The class numbers/values do not have to match the greyscale values fed to the DeepGeoMap model as they are just internal placeholder values that are later assigned back to the class when the model predicts data. For clarity table 4 shows the greyscale value for each class. Both image mosaics are then reduced by collapsing the spatial dimension, resulting in a two dimensional array (41616, 408) for the spectral training data and a one dimensional array with the according class labels (41616). The 2 arrays are then split randomly into training and test data (used for the accuracy validation) in an according 60% to 40% split. The random splitting method used keeps the class label match for the spectra (thus splits the spectra and the label pairwise). This is resulting in 4 arrays of the size (24969, 408) as xTrain data (spectral training input), (24969) as yTrain data (label training input), (16647, 408) as xTest (spectral testing input, and (16647) as yTest (label testing input). As DeepGeoMap is a data classifying (or data labelling) neural network and uses categorical cross entropy as loss function (see section 2.1.7), the label arrays (yTrain and yTest), containing the class label vector integers, are converted to binary class matrices. This results in the final binary label arrays of (24969, 16) for yTrain and for yTest (16647, 16). Then the model of DeepGeoMap reshapes the one dimensional (400,) spectral input array to (400,1) for classification training.

Hence, generally stated (also true for the following data sets), the model of Deep-GeoMap receives a random spectrum (xTrain) and the according binary class label

Table 4: Assigned greyscale values vs. sample number after Koerting (2021)

Greyscale value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Sample/Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

(yTrain), each derived from a pixel of the training input image.

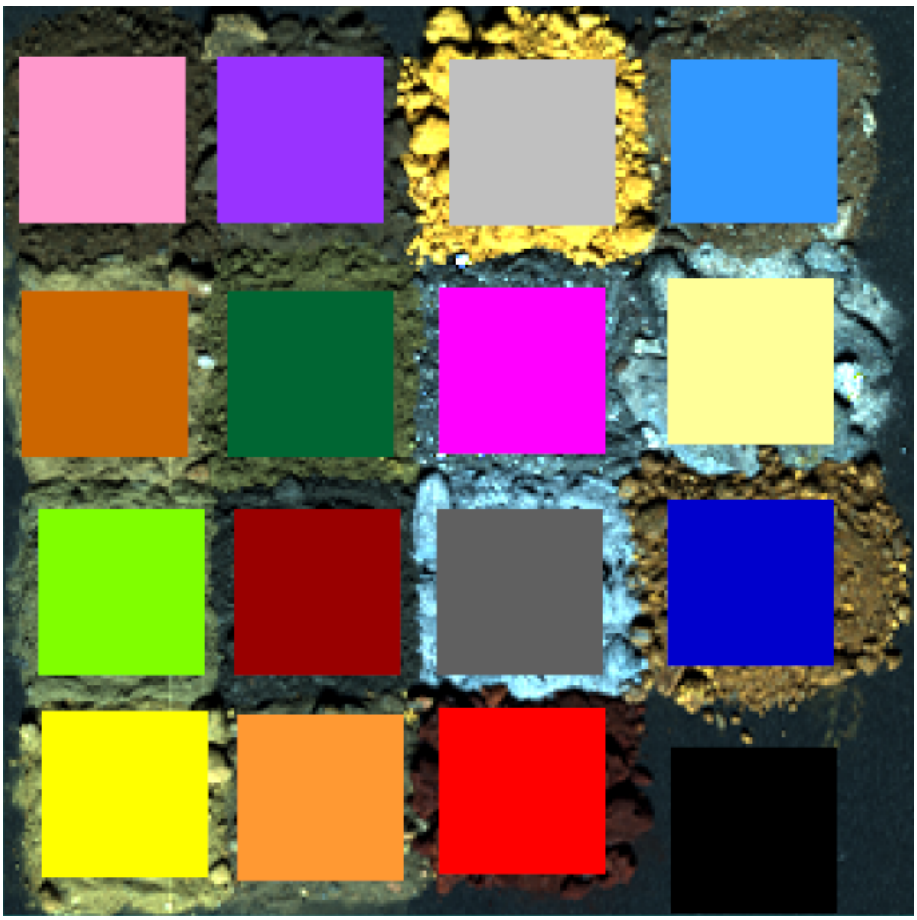

Figure 26: Brazilian iron ore samples in RGB color, recorded by the HySpex VNIR camera. The ROIs which have been used to create the training data for DeepGeoMap are displayed by the colored squares of 51x51 Pixels. Note that these areas include shadows due to the 45° illumination setup in the laboratory (see figure 13).

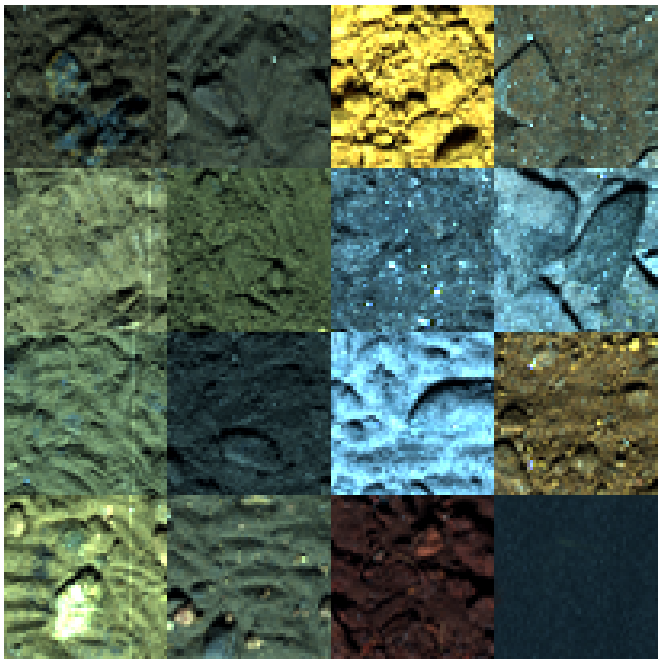

Figure 27: Mosaic of the Brazilian iron ore samples in RGB color. The Mosaic was created from 51x51 pixel image tiles selected by the ROIs shown in figure 26. The mosaic has the advantage of easily separable class boarders which ensures no training input of falsely labeled pixels.

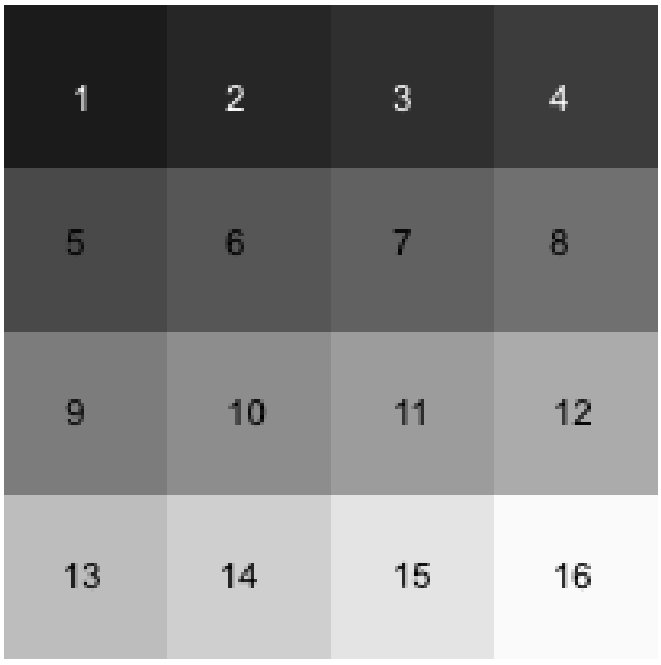

Figure 28: Label image of the Mosaic with 51x51 pixel tiles (contrast enhanced). Each of the tiles has a different grey scale value (0-15) representing each a class label of the mosaic image (figure 27). The numbers indicate the according class. The according greyscale values are shown in table 4. This is basically just a visual representation of the label classes but it enables users of DeepGeoMap to quickly check if the label image matches the training mosaic.

2.3.2 Apliki:

The Apliki sample data was pre-processed in the same manner as the Brazilian iron ore samples with adaptation to the different sample size and background. As the Apliki Lab samples are separated into 7 geochemical clusters by Koerting (2021) the ROIs and labels were chosen according to the geochemical clusters. From here on the 7 clusters will sometimes and interchangeably be referred to as classes (cluster 1 = class 1, cluster 2 = class 2). Again an additional class for the rubber and plastic background was created resulting in 8 overall classes for the training mosaic. Each ROI and hence mosaic tile has the dimensions of 41x41 pixels. Due to an unequal amount of samples per cluster the number of ROIs were also unbalanced between the cluster classes. The original image (also a mosaic) has spatial dimension of 1280x1250 pixels with 400 spectral bands (8 bad bands were removed). 12 ROIs were chosen for cluster/class 1, 10 for cluster/class 2, 8 for cluster/class 3, 4 for cluster/class 4, 24 for cluster/class 5, 26 for cluster/class 6, 2 for cluster/class 7, and 4 additional for class 0 (background material [unclassified]: cellular rubber, plastic sample boxes), resulting in 90 ROIs each of 41x41 pixel size. The ROIs are shown in figure 29. For the Apliki data also the ROIs of each class are then used to extract a spatial subset of the extent of the ROI for each class, resulting in 90 individual hyperspectral images with dimensions of 41x41 pixel containing the full spectral range of 400 bands. The resulting 90 images where then also stitched to a mosaic containing all of the images while retaining the same order as in the original image was not possible here do to the mixed arrangement of the samples. This mosaic of the 90 images from the 8 classes is shown in figure 30. The flattened mosaic of the 8 classes is then used as training data input for the DeepGeoMap model. This input is consisting of 41x41 = 1681 training pixels per tile and a total of $90 \times 1681 = 151.290$ training pixels with 400 spectral bands for the dataset. The image label mask for the Apliki mosaic is shown in figure 31. Also here both image mosaics are then reduced by collapsing the spatial dimension, resulting in a two dimensional array (151290, 408) for the spectral training data and a one dimensional array with the according class labels (151290). The 2 arrays are then randomly split 60% to 40% into training and test data. This is resulting in 4 arrays of the size (90774, 400) as xTrain data (spectral training input), (90774) as yTrain data (label training input), (60516, 400) as xTest (spectral testing input, and (60516) as yTest (label testing input). The label arrays (yTrain and yTest), containing the class vector integers, are here as well converted to binary class matrices, for the input binary label arrays of (90774, 8) for yTrain and for yTest (60516, 8). The preprocessing and model training for the Apliki field scan utilised the exact same image and procedure as described here with the only exception that 10 additional bands needed to be cut prior to training in order to let the model train in the same spectral range as the Apliki

Field data was provided (390 bands) (see section 2.2.2). Hence, only the array sizes of xTrain (90774, 390) and xTest (60516, 390) changed.

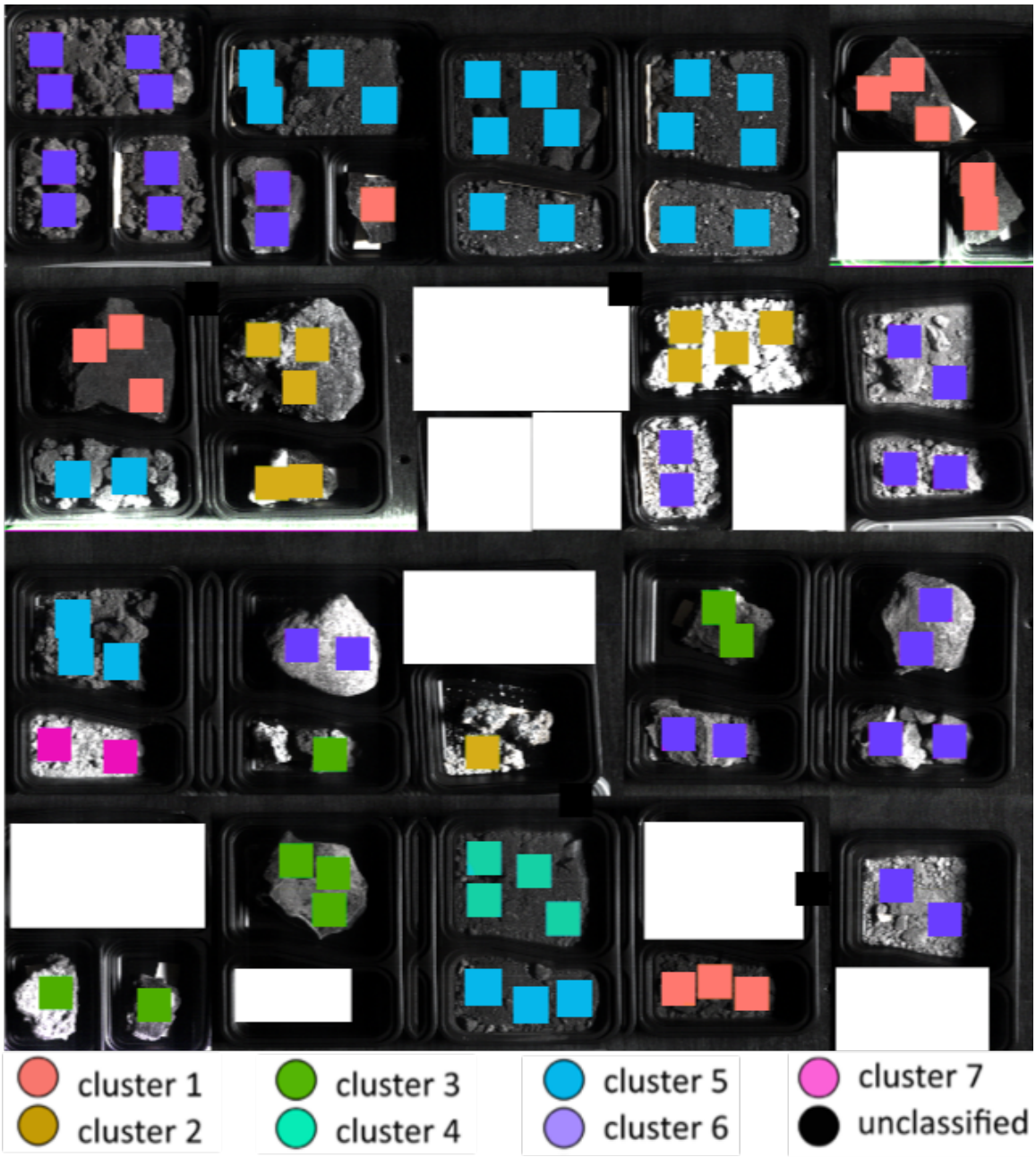

Figure 29: Apliki sample mosaic in Greyscale, recorded by the HySpex VNIR camera. The ROIs which have been used to create the training data for DeepGeoMap are displayed by the colored squares of 41x41 Pixels. Samples that were not geochemically validated were masked (white squares). The ROIs intentionally include shadows which are caused from the 45° illumination setup in the laboratory (see figure 13). The shadows were included in the ROIs to evaluate DeepGeoMap low illumination classification capability.

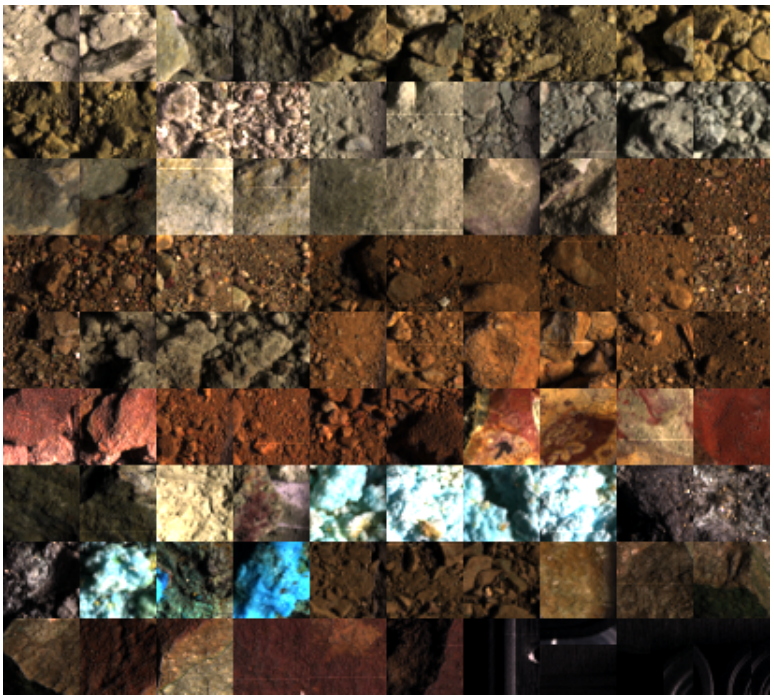

Figure 30: Mosaic of the Apliki samples in RGB color. The Mosaic was created from 41x41 pixel image tiles selected by the ROIs shown in figure 29. The mosaic has the advantage of easily separable class boarders which ensures no training input of falsely labeled pixels.

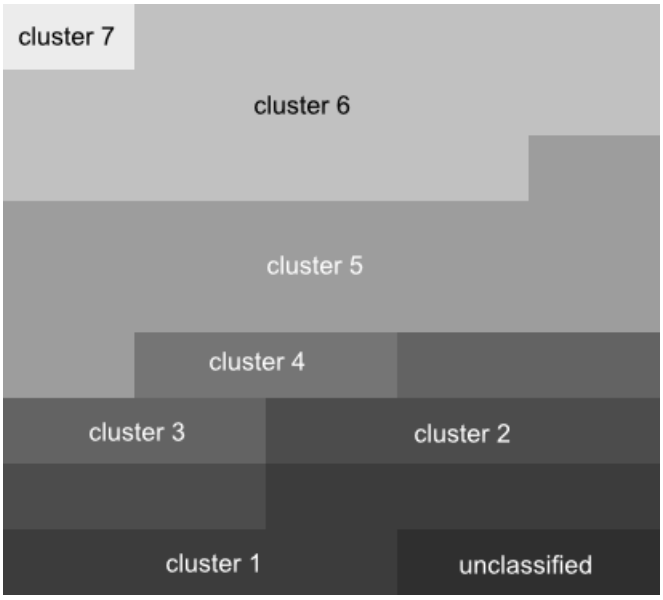

Figure 31: Label image of the mosaic with 41x41 pixel tiles (contrast enhanced). Each of the tiles has a different grey scale value (0-8) according to their respective class label of the mosaic image (figure 29 and figure 30). For better visibility/differentiability the according cluster numbers have been added.

2.3.3 Indian Pines

The Indian Pines data set does not contain any geochemically validated laboratory ground truth data that the DeepGeoMap model could be trained with. The data just consists of a hyperspectral airborne image and a ground truthed file that includes ROIs of the regions of this one hyperspectral image that have been ground truthed by classical manual ground observation. In the following the procedure is explained that was used to feed the DeepGeoMap model training data. The areas indicated by the ROIs (in Figure 32) have been cut out with ENVI and all areas not covered by the ground truth ROIs were set to 0. resulting in a training image that contained spectral data only for the ground trhuth regions.

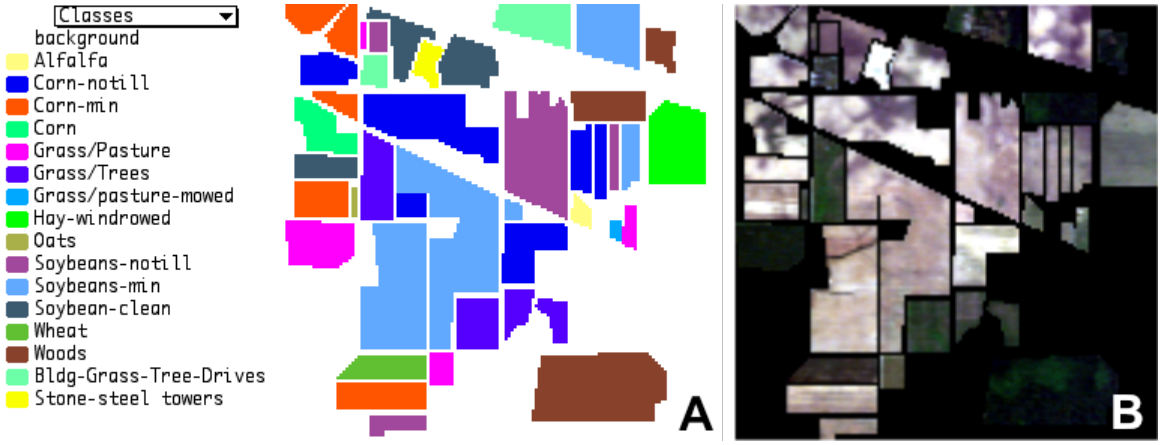

Figure 32: A; The Indian pines ground truth with classes in the original color-scheme, and B; The hyperspectral training image cut out from the original image to the dimensions of the ground truth ROIs true color image of the scene (Red: band 27 (646.7 nm), Green band 17 (547.6 nm), Blue band 7 (449.1 nm), the black area has spectral values set to 0) [6]

The Indian pines data contained 16 class labels for the images (figure 32). The black area of the training image in figure 32B is hence added as the 17th class label shown as white background in figure 32A. The image has a spatial dimension of $144 \times 140 pixels$ with 220 spectral bands each. The pre-processing from here on is the same as for the prior data sets. The spatial dimensions are flattened resulting in 21025 spectra with 220 bands as training data (including the black background pixels). The data is then also split randomly 60% to 40% into training and test data. respectively. Resulting in an xTrain array of size (12096,220), (12096,) for yTrain, (8064,220) for xTest, and (8064,) for yTest. The label arrays are then also converted to binary class matrices resulting in (12096,17) for yTrain and (8064,17) for yTest.

2.3.4 Computation

Hardware and software environment: The DeepGeoMap model training has been carried out on the following hardware environment: A 9th Generation Intel®

Core™i9-9900K processor (8 cores/16 threads), with 5.00 GHz single core turbo frequency and 16 MB Intel® Smart Cache, 128 GB of DDR4 RAM with a serial speed of 3000 MHz, a graphical processing unit (GPU) NVIDIA GeForce RTX 2080TI with 11 GB GDDR6. The following software environment was used: by Ubuntu 20.04.1 LTS x64 as operating system, CUDA 11.0, TensorFlow 2.3.1 with GPU support and Python 3.7 as programming language. This setup allowed the majority of the calculations (model training) to be carried out on the GPU, which lead to a significant performance increase vs. training through the CPU, and allowed relatively quick hyperparameter tuning (filter sizes, batch size, activation functions, learning rate etc.).

Training: A learning rate of 1×10^{-4} and a batch size of 32 (32 spectra and their according labels are seeded to the network at once) was chosen for every data set and model training. The training times varied based on the combined training and testing input size in terms of spatial / amount of spectra, and spectrally / number of bands, for each data set. For the Brazilian iron ore (41,616 pixel/spectra, 408 bands, and 3,839,056 parameter) one epoch was completed in about 11 seconds on average (with 8-11 seconds time span per epoch). For the Apliki data set which has 8 bands less (400 bands), but much more pixel (151.290 pixel/spectra, 3,838,536 parameter) one epoch took about 33 seconds on average. For the noise and water band adjusted Apliki laboratory data (151.290 pixel/spectra, and 390 bands, 3,838,536 parameter) it took as well around 33s on average per epoch. The spectrally and spatially much smaller Indian Pines data set (20160 pixel/spectra, and 220 bands, 3,839,121 parameter) required around 2s training time per epoch. Once a DeepGeoMap model is trained with sufficient epochs it can rapidly classify an hyperspectral image in few seconds to a few minutes on this system (depending on the image size). The overall amount of epochs required for each data set and the classification results a shown and discussed in the following sections 3 and 4.

Image classification: Since DeepGeoMap is a 1D network the target images that are being classified with DeepGeoMap models need to be flattened as well while storing the original spatial coordinates in variables. The model predicts the binary class for each spectra in this flattened array. The prediction results in an array with the dimension of the number of spectra and the number of binary classes. This classified binary array is then converted to the according classes in float format and the dimensions are expanded back to the according spatial domain of the original target image. Resulting in an image array with the predicted class numbers in float format. Then an RGB classification image of the same spatial extend is created that is colorized in accordance to the predicted classes found in the prediction image.

3 Results

3.1 Brazilian Iron Ore Samples

For the Brazilian iron ore images a number of different epochs have been tested. The predicted classification images of DeepGeoMap model that have been trained for 65 epochs, 80 epochs, and 500 epochs are shown in the following. Figure 33 shows the true color image of the Brazilian iron ore samples (A) and the validation colors and class numbers (B).

Figure 34 shows the predicted classification image from the DeepGeoMap model after 65 epochs. Training took around 715 seconds (about 12 minutes) with around 11 seconds per epoch. The overall classification accuracy (OA) of the model after 65 epochs is 0.9846 (98.46%) with a loss of 0.0497. The overall accuracy was calculated through the restored model weights on complete reference data covered by the rois shown in Figure 26. The validation accuracy (VA), which is calculated just with data that the model has not been trained with (xTest, YTest), is as well 98.46% and validation loss is 0.0492. The average accuracy ($AA = (2 \times OA + k - 2)/2$, where OA = overall accuracy and k = number of classes) is 99.81%. By visual inspection and comparison to the validation image (figure 33) the predicted image shows that the vast majority of pixels are mapped correctly according to their sample class location. Sample classes 1, 3, 5, 6, 7, 9, 10, 11, 12, and 15 show almost no misclassified pixels with the exception of a few. Sample class 2 shows some misclassification at the top part of the sample, a misclassified rock at the top as class/sample 14, and a few pixels falsely classified as sample class 15. Sample 4, 8 and 13 also show minor misclassifications in very dark/shadowed areas if visually cross-checked with figure 33A.

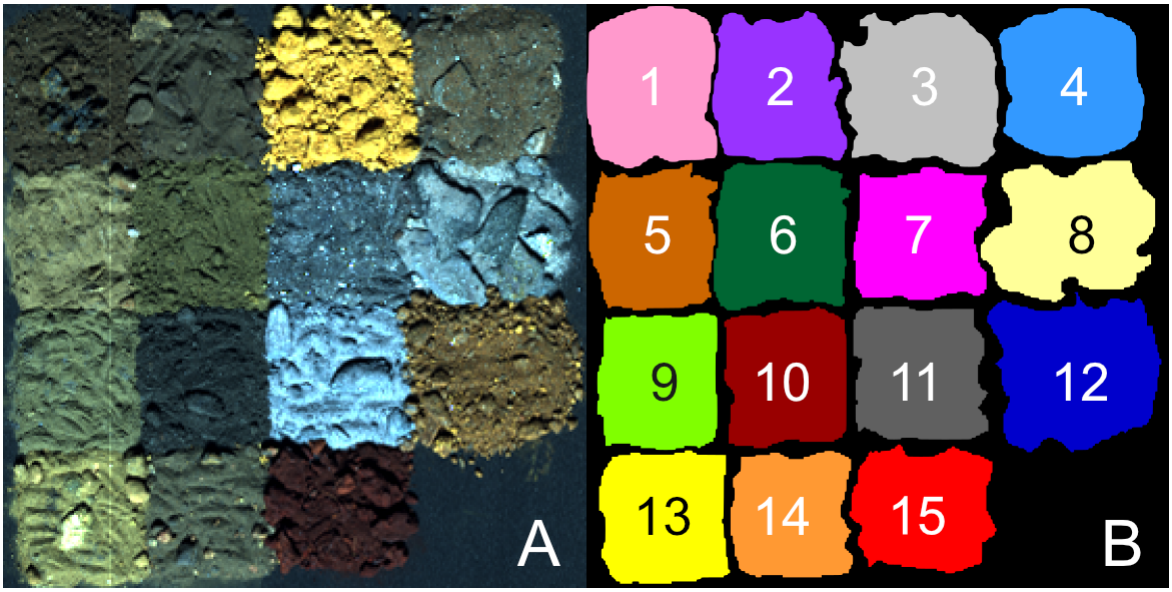

Figure 33: A: True color image of the samples. B: This image shows the classification validation modified after Koerting (2021). The class/sample number are marked on the according color validation.

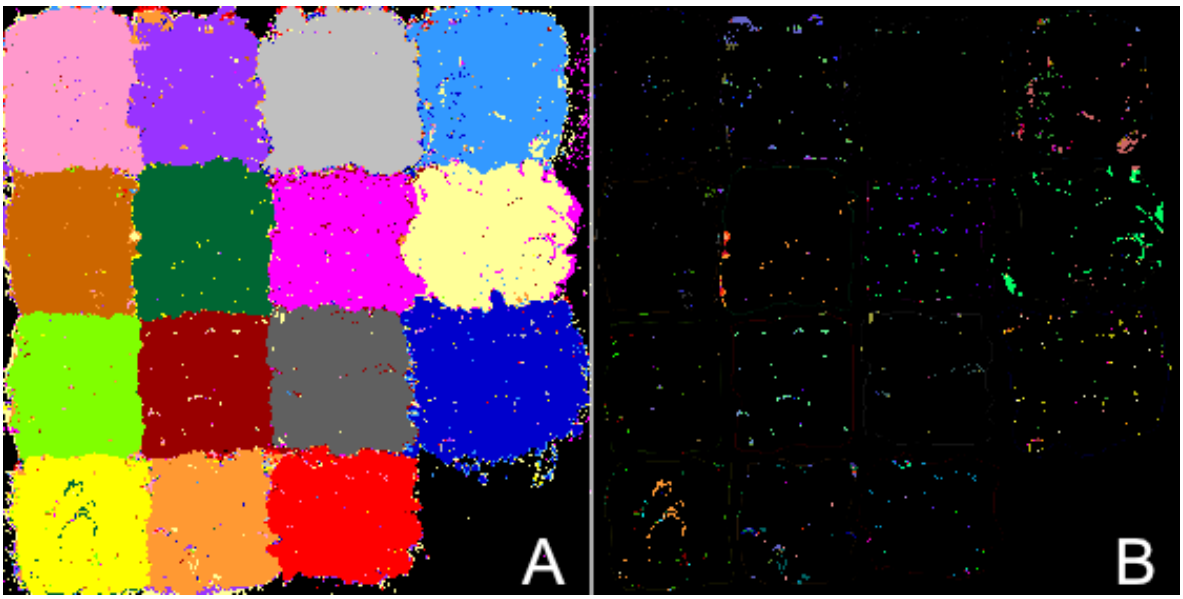

Figure 34: A: This image shows the predicted classification of the DeepGeoMap model after 65 epochs. B: False classifications/error map. Pixels that do not match the validation ROIs shown in figure 33 are shown here. The colors of the mismatching pixels are the result of RGB value subtraction of the validation image from the predicted classification image. Areas outside of the validation ROIs hence are not considered.

Training was then set for 80 epochs. Figure 35 shows the predicted classification image from the DeepGeoMap model after 80 epochs. Training took around 880 seconds (15 minutes) with around 11 seconds per epoch. The overall classification accuracy (OA) of the model after 80 epochs is 0.9846 (98.93%) with a loss of 0.0393. The validation accuracy (VA), which is calculated just with data that the model has not been trained with (xTest, YTest), is 98.43% and validation loss is 0.0638. The average accuracy (AA) is 99.86%. By visual inspection and comparison to the validation image

(figure 33) and the prediction image of the 65 epoch model it can be stated that in some areas a slight improvements are visible, while other falsely predicted areas have gained some more clustered falsely classified pixels. Still the predicted image shows that the vast majority of pixels are mapped correctly according to their sample class location. The Sample classes 1, 3, 5, 6, 7, 9, 10, 11, 12, and 15, that were already quite well classified by the 65 epoch model even show slight improvements. Sample class 2 still shows the misclassified rock at the top part of the sample. Samples 4, 6, and 8 show larger improvements even in the very dark/shadowed areas. Sample 7 and 13 show more misclassifications than in the 65 epoch model image.

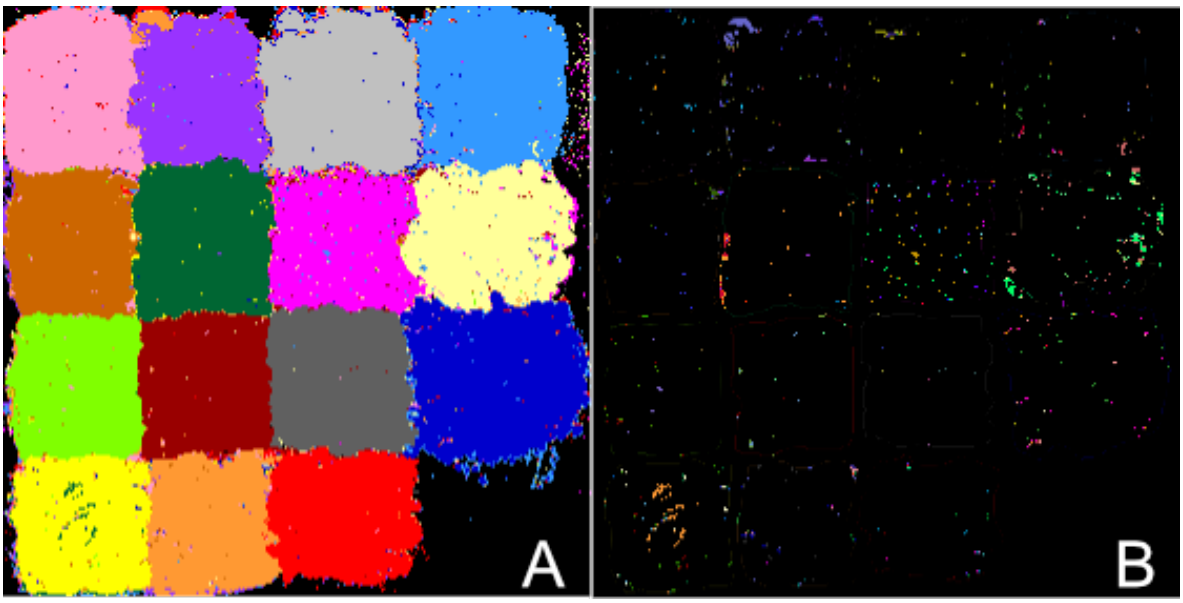

Figure 35: A: This image shows the predicted classification of the DeepGeoMap model after 80 epochs. B: False classifications/error map. Pixels that do not match the validation ROIs shown in figure 33 are shown here. The colors of the mismatching pixels are the result of RGB value subtraction of the validation image from the predicted classification image.

A third run with 500 epochs is as well presented. To train the model with this many epochs was done in order to analyse overfitting. The general approach in neural network model training is to let a model run until overfitting is detected. The basic strategy of this approach and the overfitting issues are discussed in section 4.1. The training took around 5500 seconds (about 92 minutes) with around 11 seconds per epoch. The overall classification accuracy (OA) of the model after 500 epochs is 0.9960 (99.60%) with loss of 0.0392. The validation accuracy (VA), which is calculated just with data that the model has not been trained with (xTest, yTest), is 99.58% and validation loss is 0.0605. The Average accuracy (AA) is 99.95%. These are very high accuracy results for the validated areas shown in figure 26. The visual inspection of the DeepGeoMap classification image is at least in the most parts of the image consistent with these numbers (figure 36). The image shows very few misclassifications in the middle parts of the samples, except for sample classes 7, 8 and 13. The classifications

of samples 7 and 13 still show a few larger misclassifications around the shadowed areas (figure 33). The wrongly classified pixels are more clustered and less spread as in the models trained with 65 or 80 epochs.

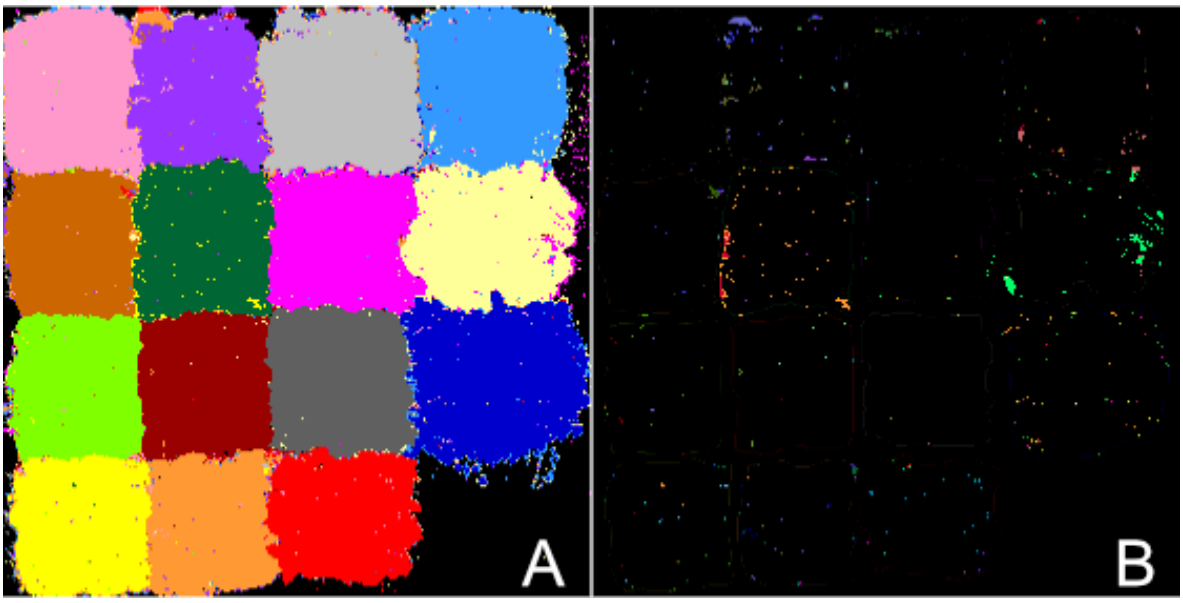

Figure 36: A: This image shows the predicted classification of the DeepGeoMap model after 500 epochs. B: False classifications/error map. Pixels that do not match the validation ROIs (figure 33).

3.2 Apliki

The classification results for a model that has been trained for 42 epochs are presented here. One epoch took around 33 seconds on average, resulting in a model training time of 1386 seconds (around 23 minutes) for 42 epochs. The Overall Accuracy (OA), was calculated through the area of the ROIs shown in figure 29, and not the ROIs shown in figure 37. This is important to note as the OA for the whole set is likely lower. The OA for the input data of the 42 epochs model reached 99.32% with a loss of 0.0222. Hence the average accuracy (AA) for the 8 classes (including background) is 99.83% (for the ROIs in figure 29. Hence the much more important and representative metric here is the VA. It reached 98.83% with a validation loss of 0.0382. The VA also correlates more to the visual appearance of the classification map shown in figure 37 C and error map (D). Visual inspection shows that the overall classification result seems very accurate for the areas shown in the validation image (figure 37 B), with the exception of the black plastic and rubber background. Especially the plastic containers in which the samples were placed reflected light around the samples that is often misclassified by the DeepGeoMap model.

For the classification of the Apliki mince face scan, a DeepGeoMap model was used that was trained with the hyperspectral laboratory images of the collected samples (see section2.2.3 and 2.3.2). The model was therefore trained for 42 epochs with an 390

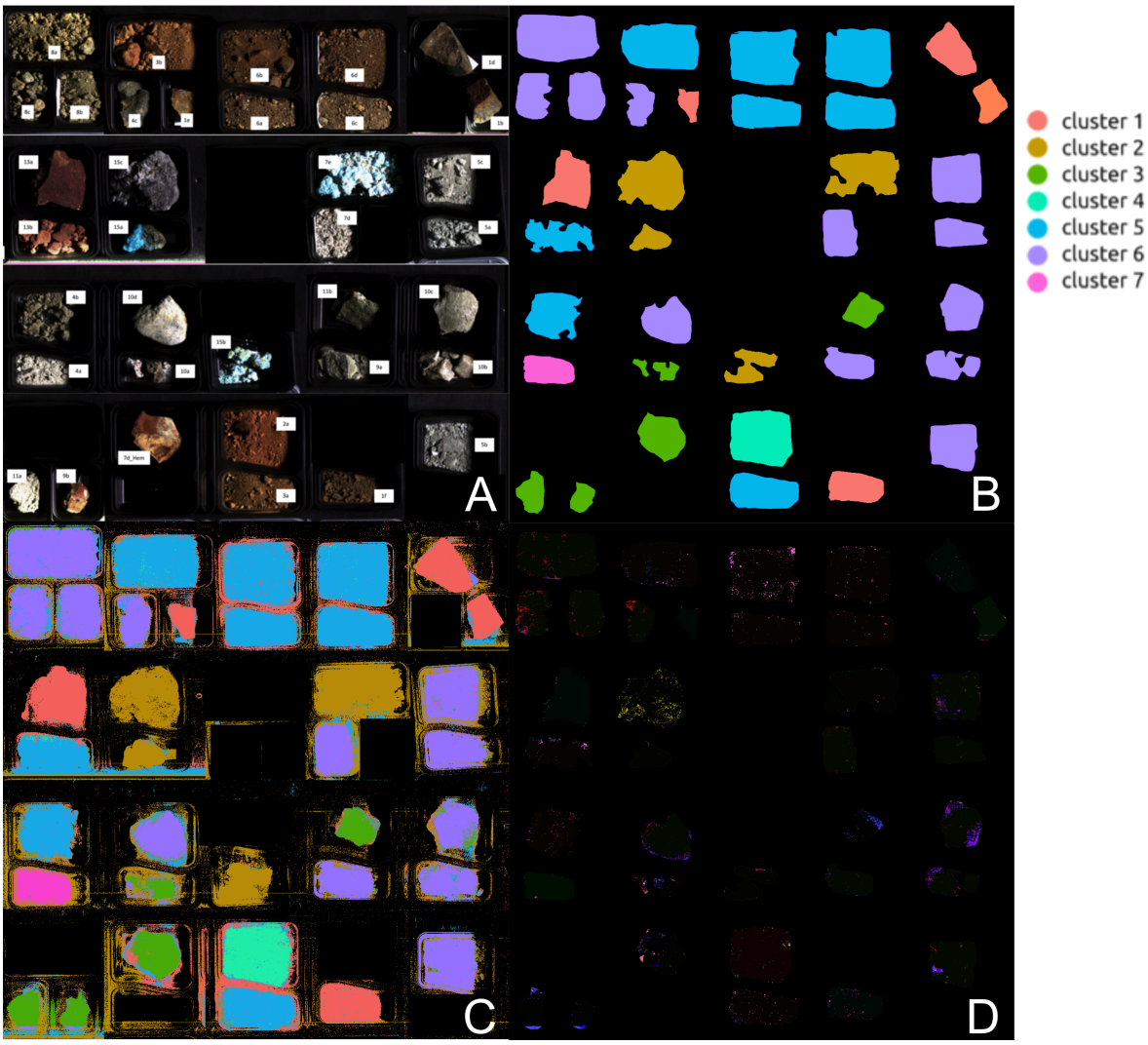

Figure 37: A: Apliki mine samples hyperspectral mosaic of 9 images stiched together, True color (R: 640nm – band 63, G: 549nm – band 38, B: 458nm – band 13); B: Sample cluster/class validation. C: DeepGeoMap classification after 42 epochs, with a VA of 98.83%. D: Shows the misclassifications/error map of of the samples of the DeepGeoMap prediction (excluding the background misclassifications). The colors are again like in the above section the result of RGB color subtraction -miss-matching classes are hence highlighted. Images A and B: Koerting (2021) [24].

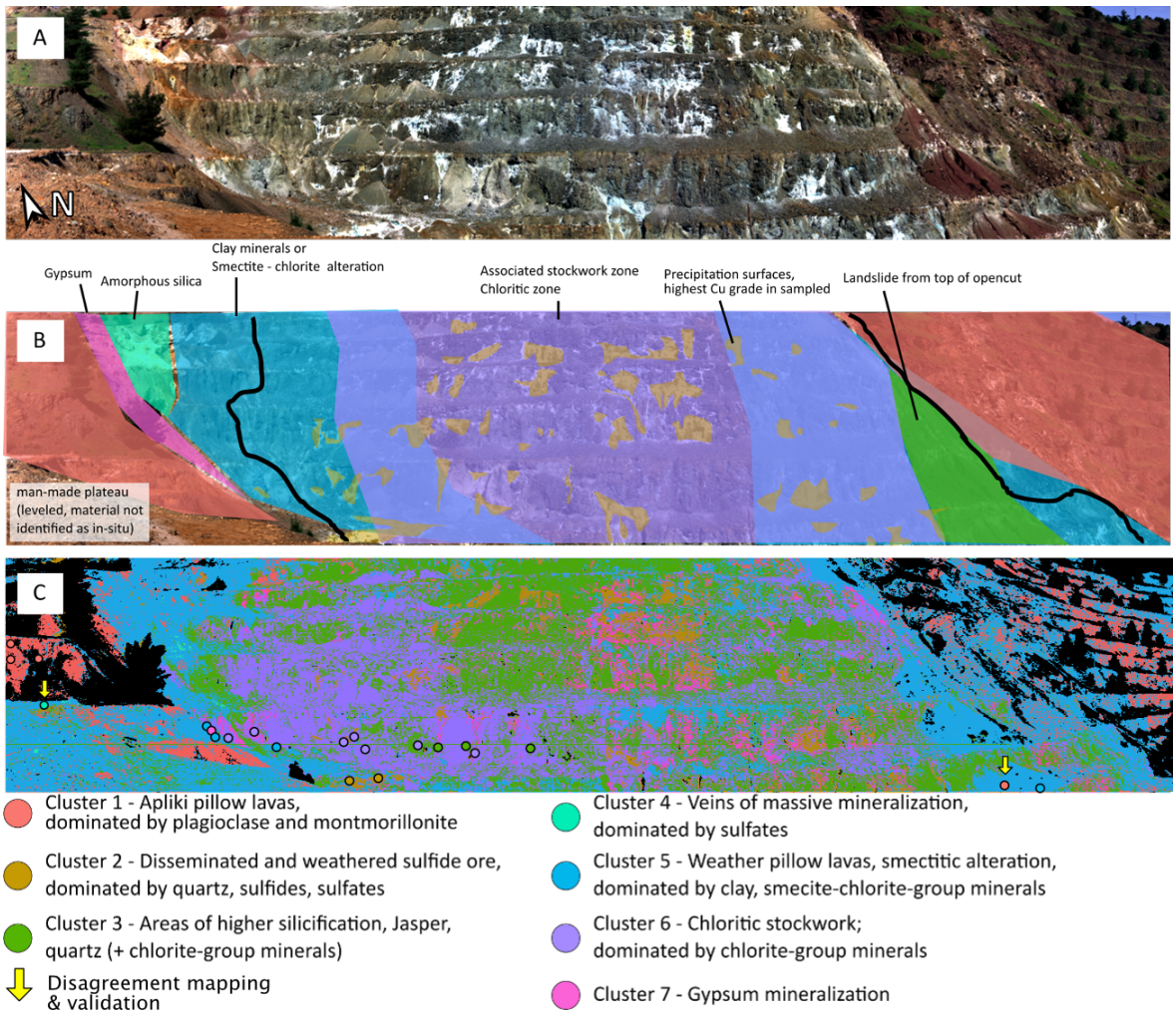

Figure 38: A: "RGB representation of the Apliki open cut", True color (R: 640nm – band 63, G: 549nm – band 38, B: 458nm – band 13); B: "the geological mapping of the open cut (B), based on Antivachis (2015)" C: Classification by a DeepGeoMap model trained for 42 epochs on the Apliki laboratory samples, 20 out of 22 sample points fit the classification map (see figure 37). Legend (modified), text, and the images A and B after Koerting (2021)

band version of the same laboratory samples mosaic shown in figure 37. The only adaption that had to be done was to exclude the noisy bad bands the same way as for the mine face field scan. This is necessary as a DeepGeoMap model that has been trained in a certain spectral domain can only classify other image in the same spectral domain it has been trained in. The validation accuracies of the 400 and 390 band laboratory image classification were almost identical (98.83% vs. 98.41% val. accuracy respectively). Hence, cutting very noisy bands does not impact the accuracy significantly. The resulting classification is shown in figure 38. The actual accuracy for a real mine face scan cannot just be calculated easily, it has to be analysed with geological expertise in order to determine if the classifications are sensible. Hence, the basis for the following analysis is the dissertation of Koerting (2021) and the publication by Antivachis (2015) (see [24, 4]. The area in the very west of the open cut is correctly mapped as Cluster 1 (pillow lavas). The gypsum (cluster 7) and Amorphous silica (cluster 4) is not mapped in the right area in abundance rather very small scattered patches where classified by the DeepGeoMap model. The western and eastern parts of the image scan are correctly mapped as cluster 1 (red), related to the pillow lava occurrence. The small landslide in the east of the image is mapped as cluster 5 and 3 by the DeepGeoMap model, which in parts correlates with the analysis of Antivachis (2015). It also confirms the samples collected in the area, which where classified as either cluster 1, or cluster 5 by Koerting (2021). The middle part in the open cut stockwork related zone (Cluster 6 and associated mineralogy) is mapped in the majority a little too far to the left in the DeepGeoMap classification, but overall still confirming the general position. The majority of the Cu-rich precipitates (weathered sulfide ore, cluster 2 and yellow) is mapped at the right positions when compared to the geological interpretation (B), but could be more pronounced to fully comply with the map from Antivachis (2015). Many pixels in the centre are classified as Cluster 3 too often to comply with the geological map (see section 4.2). One metric available that can be expressed in numbers are the number of sample locations in which the classification of the mine face scan agrees with the sample geochemistry/cluster, where the DeepGeoMap classification agrees with 20 of 22 sampling locations.

3.3 Indian Pines Data Set

The DeepGeoMap model was trained for 165 epochs using 60% of the validation data as training input, including the background: 12096 pixels/spectra, and excluding the background with 6171 pixel/spectra. Table 5 list the complete training samples per class. The OA accuracy reached is 96.21%, the AA 99.55 % and the VA 96.04%. Note that the average accuracy is so high because the training data includes the background as the largest sample class (with value 0, which is simple to train). This is why the

VA is the most reliable metric in this case. Over 165 epoch signs of overfitting started to appear as the VA did not rise any more. Figure 39 C show the classification result of the 165 epoch DeepGeoMap model of the Indian pines scene. Figure 39 D shows the classification of the validation image, hence showing the misclassifications which impact the VA.

Figure 39: A; The hyperspectral training image cut out from the original image to the dimensions of the ground truth ROIs true color image of the scene (Red: band 27 (646.7 nm), Green band 17 (547.6 nm), Blue band 7 (449.1 nm). B; The Indian pines ground truth with classes in the original color-scheme, [6]. C: DeepGeoMap 165 epoch model classification of the Indian Pines scene with an overall accuracy of 96.21%. D: DeepGeoMap 165 epoch validation image classification showing the misclassifications in each validation ROI. The VA is 96.04%.

Table 5: The number of training and test samples for Indian Pines dataset used for training the DeepGeoMap model.

No.	Class	Samples	Train	Test
0	Background	9794	5865	3929
1	Alfalfa	54	29	25
2	Corn-notill	1434	851	583
3	Corn-mintill	834	489	345
4	Corn	234	147	87
5	Grass-pasture	497	306	191
6	Grass-trees	747	431	316
7	Grass-pasture-mowed	26	13	13
8	Hay-windrowed	489	301	188
9	Oats	20	12	8
10	Soybean-notill	968	592	376
11	Soybean-mintill	2468	1485	983
12	Soybean-clean	614	373	241
13	Wheat	212	139	73
14	Woods	1294	791	503
15	Buildings-Grass-Trees-Drives	380	22	358
16	Stone-Steel-Towers	95	47	48
Total		**20160**	**11893**	**8267**

Table 6: DeepGeoMap model overview for the Indian Pines data set.

Model	**DeepGeoMap**
Epochs	165
Training sample	0.6
Selected samples Including Background	11893
Sampling mode	Random
Patch Size	-
Learning Rate (LR)	0.001
Batch size	32
Training time [s]	338
Classification time [s]	>1
Average accuracy (AA) [%]	**98.43**
Overall accuracy (OA) [%]	**96.21**
Validation accuracy (VA) [%]	**96.04**

4 Discussion

4.1 Brazilian Iron Ore Samples

The DeepGeoMap models classified the Brazilian iron ore samples with very high accuracies (65 epochs: 98.46%, 80 epochs: 98.93%, and 500 epochs: 99.60%). First of all it has to be said that the models were trained with a significant amount of data, 24969 pixels/spectra which represent 60% of the square shaped input ROIs seen in figure 26 and around 32% of the overall image pixels (78400 pixel). This although is generally an important aspect of this type of 1D neural network. The ability be trained with large amounts of laboratory data without any spatial context and use these trained models on images of different resolutions for example. Also, DeepGeoMap does not easily overfit from large amounts of training pixels from spatially close sample inputs as the data is randomly shuffled and DeepGeoMap is anyway not training any spectral-spatial relationships, and or shapes found in the sample data. The assumption that the model overfits by training 500 epochs, has least a little backup by the calculated accuracies and loss. Usually overfitting, is in machine learning defined as, and occurs when the accuracy continues to rise during training while the validation accuracy and validation loss stays the same or gets worse, hence the model loses in generalization ability [13, 7]. For 65 epochs of training the model showed a validation accuracy (VA) 0f 98.46% and validation loss (VL) of 0.0492, for 80 epochs VA of 98.43% and VL of 0.0638 and for 500 epochs VA of 99.58% and VL of 0.0605. So the VL stayed the same over the course of 15 more epochs until the 80th epoch, but reached a much better value after training 500 epochs, while the loss slightly grew. So it is hard to say if the model is actually overfitting the data until 500 epochs, latest research by Belkin et al. (2019) shows that complex models, such as from deep learning neural networks, perform at high accuracy in spite of being trained to exactly fit or interpolate the data [7]. IBM (2021) states that this finding is in direct contradiction with the (historical) literature on this topic [13]. So maybe overfitting occurs at over 500 epochs, but very clearly not before the 65th and 80th epoch. The border parts of the samples, which are not in any of the train (xTrain) or test data (xTest), do show a visual improvement in the classification between the 65 epoch and the 500 epoch model. If the model would have overfitted these areas should get much worse classification results. Overfitting causes the network to loose its generalization ability and hence can only identify the pixels better that where in the training input (xTrain) while non training input pixels (xTest and all the others) get classified wrong more often, as the network never 'sees' the data during training. The overall classification of the 500 epoch model show the least scattered misclassified pixels. So overfitting is very unlikely the case here, hence the model basically just learned to recognise the spectra in the data better. To fully back

this up, however, more samples of the same classes and geochemistries are needed to be analysed. Considering the geological and geochemical background of the samples, that all iron ores from the QF are inter-gradational and just represent different stages of weathering ([24], section 2.2.1), it could also be argued that the neural network is able to see beginning of weathering transitions to the other ore grade types. This interpretation, although possible, would require additional data and geochemical analysis of the sample parts that have been classified differently form the majority of the samples to be justifiable. Nevertheless the general classification capabilities are very high especially when compared to other methods that classified this data set, which are thoroughly compared by Koerting (2021). In the following a few of theses results will be compared to the DeepGeoMap results and discussed.

Figure 45, shows the results of DeepGeoMap (A) and three different classification algorithms/methods used by Koerting (2021) for the Brazilian iron ore Image. All of the algorithms/methods deployed used shadow masked (shadow spectra set to value 0) version of the hyperspectral image for analysis. Figure 45 B shows the classification image of the EnGeoMap 2.1 algorithm with is an improved version of the EnGeoMap 2.0 version both developed by Christian Mielke [24, 38]. The algorithm uses a spectral library consisting of 5x5 pixel average spectra acquired form the 15 samples. [26]. EnGeoMap 2.1 achieved an accuracy of 53.32% for this data set. Figure 45 C shows the classification result of the classification result of a GFZ intern Binary Feature Fitting algorithm (BFF), which was also developed by Christian Mielke, is utilising a 42 band spectrally downsampled version of the image and also a spectral library consisting of 5x5 pixel average spectra, it achieves an accuracy of 74.35%. Figure 45 D shows the classification result of the ENVI Spectral Angle Mapper (SAM) achieving 65.95% accuracy also using a spectral library. Comparing the classification images/maps from these methods with the DeepGeoMap model classification shows that it performs very well in terms of accuracy. Most of the samples classes show much more misclassifications compared to the DeepGeoMap classification, with the exception of the BFF classification of sample class 3 which show almost no misclassification. The visual comparison of the classifications although has the constraint that the methods used by Koerting (2021) where slightly different as shadow masked versions of the image were used for classification. For this thesis no shadow masks where used to test the classification capability of DeepGeoMap models in poorly lit areas. From figures 34, 35, and 36 it can be seen that the shadowed areas where classified well and that the accuracy for them increases with the number of epochs. At epoch 500 only sample class 8 has a small misclassified shadowed area. This ability of DeepGeoMap to map the correct sample class under bad lighting conditions can bee seen as an advantage over the classical algorithms presented here. Nevertheless the classical algorithms just used one averaged spectra per sample class to achieve the classification shown in fig-

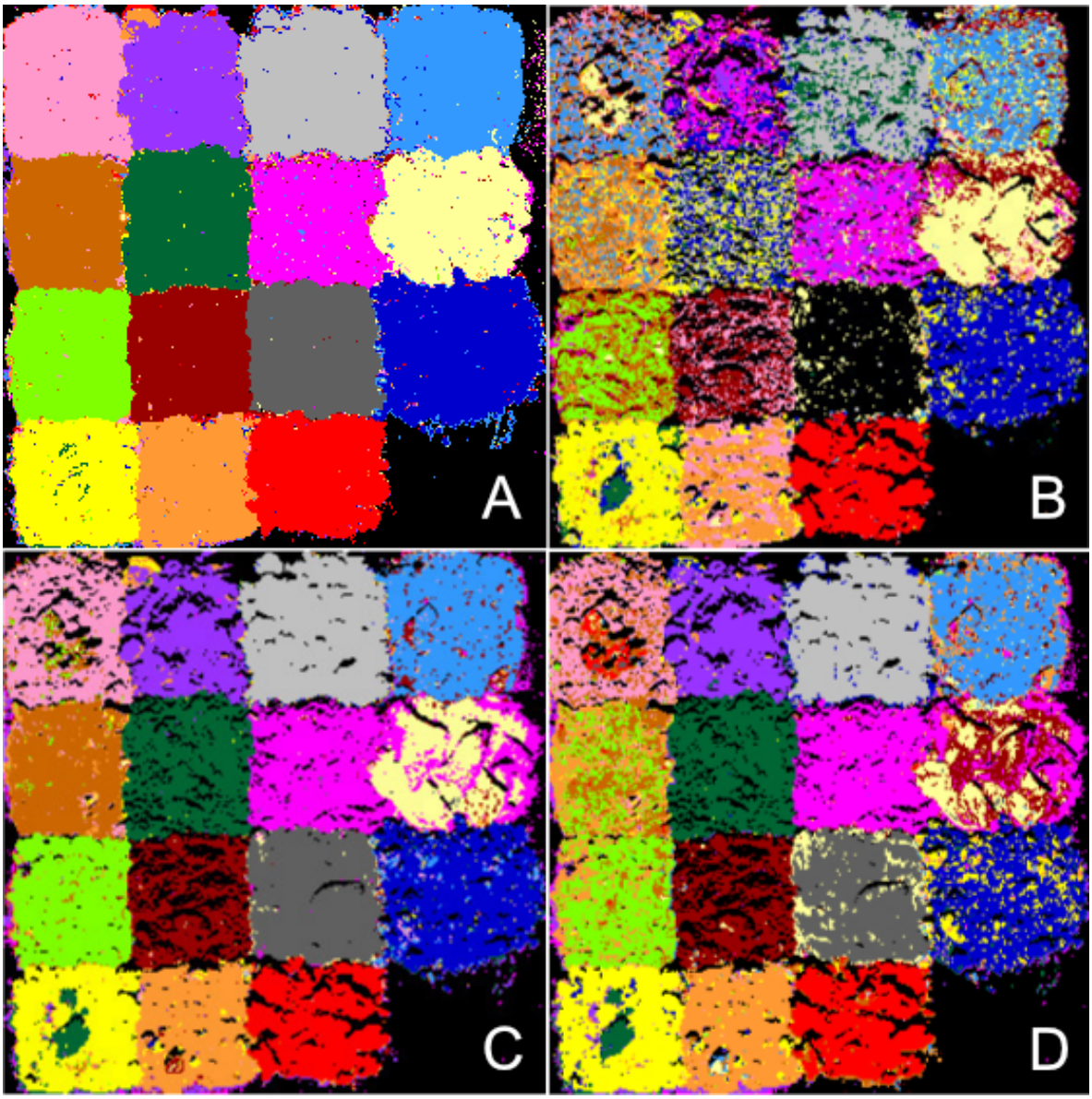

Figure 40: Overview of different classification methods for comparison of DeepGeoMaps classification capability. A: This image shows the predicted classification of the DeepGeoMap model after 80 epochs using the full 408 bands, resulting in a OA of 98.93%; B: Shows the shadow masked classification result of EnGeoMap 2.1 using the full 408 bands with a classification accuracy of 53.32%; C: Shows the shadow masked classification result of a GFZ Binary Feature Fitting algorithm (BFF) which is utilising a 42 band spectrally downsampled version of the image with an accuracy of 74.35%; D: shows a full 408 band shadow masked classification of the ENVI Spectral Angle Mapper (SAM) achieving 65.95% accuracy. Images B, C, and D from Koerting (2021),

ure 45, which is an ability that DeepGeoMap lacks. In general are neural networks very data demanding for training, which is an issue many researchers are improving through the spatial domain for example by implementing Markov random fields e.g. Cao et al., (2017) or Sun et al. (2015), or other methods by e.g. D'souza et al., (2020). Koerting (2021) also analyzed the computational times of the classical methods shown figure 45. EnGeoMap took 808.23 seconds, BFF, 422.59 seconds and SAM took 5.00 seconds to compute the classification although DeepGeoMap took around 880 seconds for training 80 epochs, this is not directly comparable. While DeepGeoMaps model training time is 880 seconds for 80 epochs, this model (after being trained) can then classify any hyperspectral image of the size and spectral domain of the Brazilian iron ore samples in under 10s. Also the largest hindering fact for direct comparison of computational times that a different computer setup was used for the classifications. Also the training weights of TensorFlow neural network models can be saved after training, stored, and reloaded when needed to classify other images with them. Through storing and reloading they don't loose any of their classification capabilities for the classes they have been trained with. This can be seen as a very strong advantage of neural networks in terms of time and energy efficiency for the classification of a large amounts of hyperspectral images with pre-defined target classes, like it is the case with specific (iron) ore types, compared to classical algorithms. The 65, 80, and 500 epoch models for example were saved and can be reloaded and used to classify any new or other hyperspectral image (in the same spectral domain) that includes any of the training ore types, further the prediction/classification time would then be the same for all 3 models (under 10s). Hence investing more time in model training, by training for more epochs, can be beneficial under certain circumstances, and if the models accuracy improves through it. Koerting (2021) also used different neural networks to classify the Brazilian iron ore, three of them are in following used for comparison. A 1D Baseline NN by Audebert et al., (2019), a 1D CNN by Hu et al., (2015) and a 3D CNN by Li et al., (2017). The resulting classification maps are compared to the DeepGeoMap 80 epoch model in Figure 42, note that the colormap of the classifications is a different. A validation image with this colormap is shown in Figure 41. All of the models of the three neural networks where trained with 20858 input pixel samples (xTrain)[24] for 100 epochs with batch sizes of 100. The 1D Baseline NN by Audebert et al., (2019) achieves an accuracy of 89.8% (Figure 42 B), it shows many misclassifications for samples 1, 2, 3, 6, 8, 13, 14 and 15, while the rest showes little mismatches. The 1DCNN by Hu et al. (2015) reaches an accuracy of 75.2% and also shows the most misclassifications of all three neural networks for this data set across the samples except for sample 11 and 15, which show optically good results (Figure 42 C). The 3D CNN by Li et al. (2017), is achieving 93.9% accuracy and is hence the closest in terms of accuracy to the DeepGeoMap model. Optical inspection although shows that there are several

misclassifications in sample 1 and 9 (Figure 42 D). Koerting (2021) stated that this was very likely caused by overfitting. This seems plausible as the optical inspection of the classification does not correlate with the very high 93.9% accuracy. 3D neural networks usually need less epochs and data as 1D networks in order to reach their optimal accuracy and prediction result. The 3DCNN model as well as both 1D models where given disjoint data inputs (hyperspectral pixels with close spatial relations / or hyperspectral pixel blocks). DeepGeoMaps data on the other hand is also taken from a disjoint part of the image (see figure 26), but then splitted and randomized 60/40 in train and test data before being given to the model. For 1D models, that analyze the spectral domain only, random or disjoint input should in theory not influence the accuracy strongly, as they don't utilise the spectral-spatial relationships, with the exception of spatially clustered spectral differences inside a sample class. This is because the model would then never 'see' the spectra that are for e.g. in a corner not covered by the disjoint training input [5]. For the 3DCNN model it should make a difference as the spectral-spatial relationships are important for their accuracy and evenly distributed random input allows the model to predict the spatial shapes of the features (including grain sizes, textures for this ore data set for e.g.) in the images easier [5]. Tables 7 and 8 shows a comparison of the most important values.

Overall, the classification accuracy of the DeepGeoMap model is the highest of the four neural networks shown in figure 42, while using a little more training data (24969 pixels/spectra vs. 20858 pixels/spectra), but less epochs (80 epochs vs. 100 epochs). This was hoped for as DeepGeoMap was specifically developed for this kind of geological hyperspectral image classification (section 9). The iron ores present little spatial feature variability in the small grained parts of the samples at this resolution (grain sizes), while the rocks show high and complex variability among their spatial shapes, there are also no larger scale characteristic spatial shapes (all samples were arranged in the same 4x4cm squares) and hence classification is presumably benefitting the most from the spectral domain.

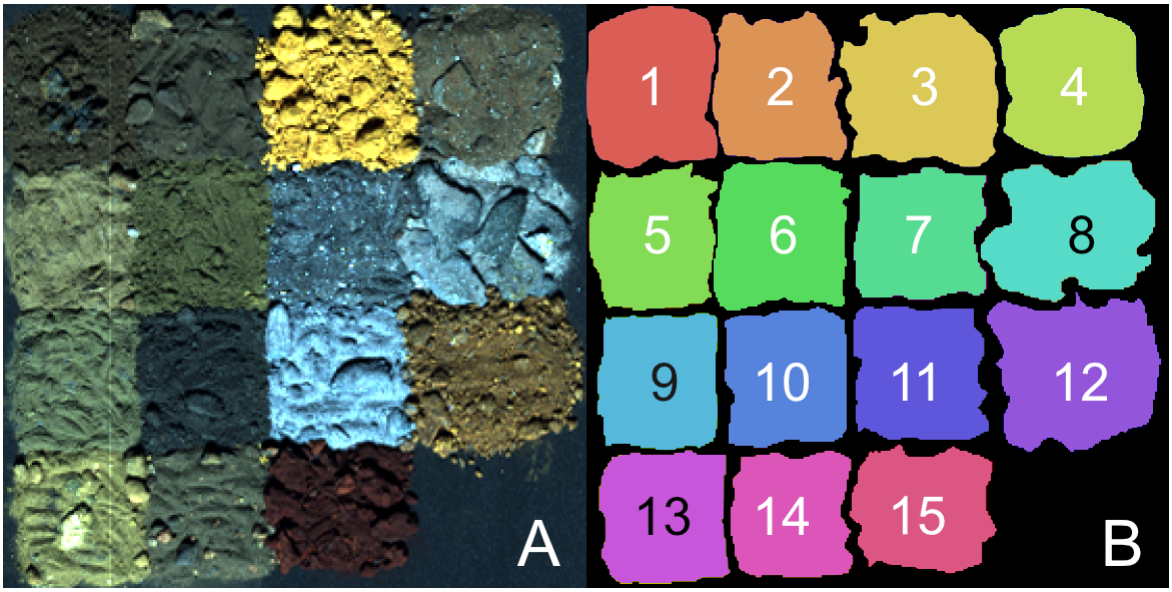

Figure 41: A: True color image of the samples. B: This image shows the classification validation with the colormap used for the Deep learning models in Figure 42, modified after Koerting (2021). The class/sample number are marked on the according color validation.

Table 7: Comparison of accuracy and computation times of DeepGeoMap and other methods for the Brazilian iron ore data set, modified after Koerting (2021). *Different computational setup.

Method (Classical)	**Computation time [s]**	**Accuracy [%]**
EnGeoMap 2.1	808.23	53.32
Binary Feature Fitting	422.59	74.36
SAM	5.00	65.95
Method (Neural Network)	**Model training time [s]**	**Accuracy [%]**
1D Baseline NN (100 epochs)	5432.47	89.80
1D CNN (100 epochs)	347.58	75.20
3D CNN (100 epochs)	5318.44	93.90
DeepGeoMap (80 epochs)	~880*	98.43
Method (Neural Network)	**Prediction time [s]**	**Accuracy [%]**
DeepGeoMap (80 epochs)	~10s*	98.43

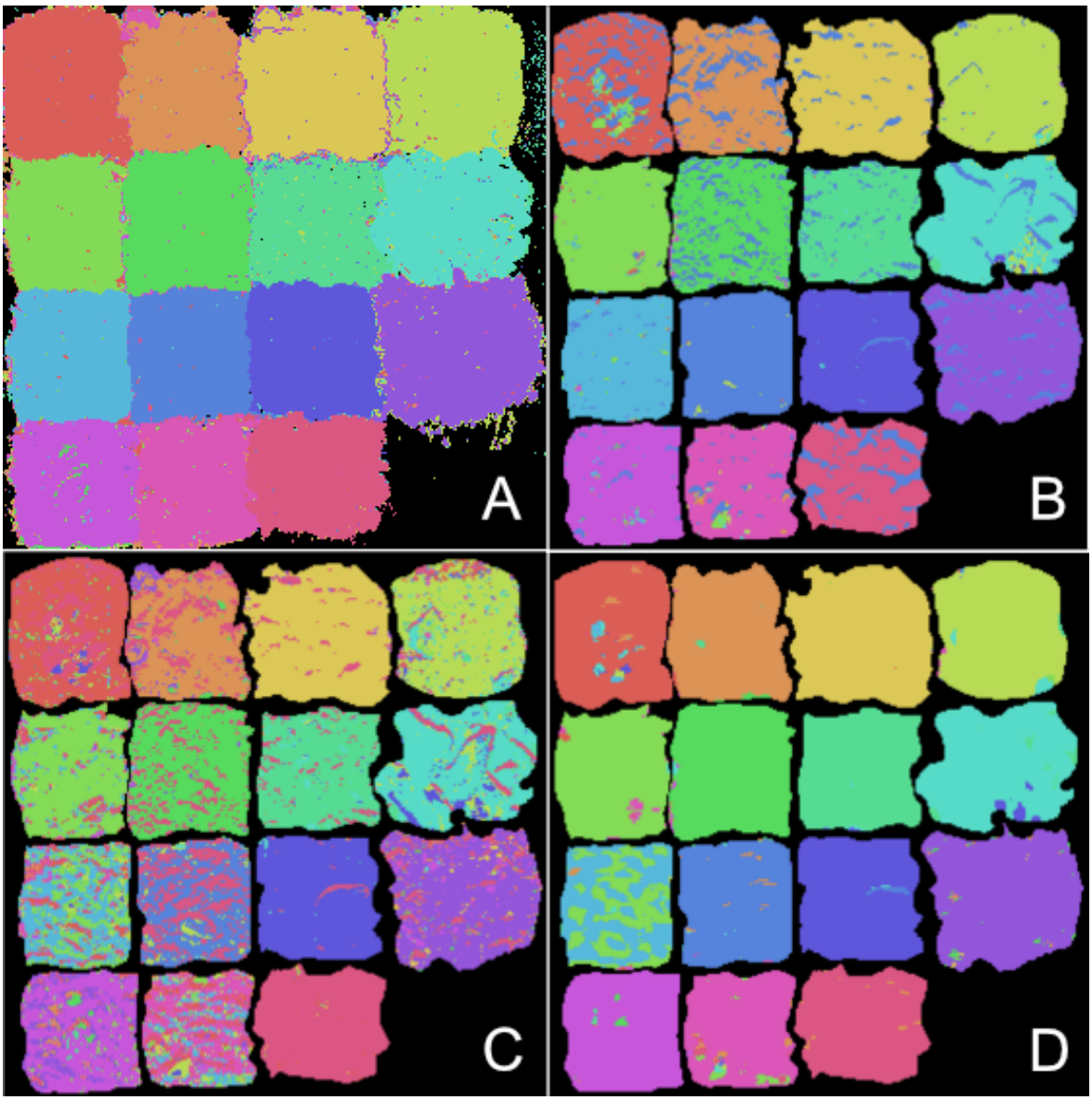

Figure 42: Overview of different classification methods for comparison of DeepGeoMaps classification capability. A: This image shows the predicted classification of the DeepGeoMap model after 80 epochs, resulting in a OA of 98.93%; B: Shows classification of a 1D Baseline NN by Audebert et al. (2019), with a classification accuracy of 89.8%; C: Shows the classification result of a 1DCNN by Hu et al. (2015), with an accuracy of 75.2%; D: shows a classification of a 3D CNN by Li et al. (2017), achieving 93.9% accuracy. Images B, C, and D from Koerting (2021)

Table 8: Comparison of accuracy and other parameters of DeepGeoMap and a 1D Baseline NN by Audebert et al., (2019), a 1D CNN by Hu et al., (2015) and a 3D CNN by Li et al., (2017) for the Brazilian iron ore data set, modified after Koerting (2021).

Model	1D Baseline NN	1D CNN	3D CNN	DeepGeoMap
Based on	(Audebert et al., 2019)	(Hu et al., 2015)	(Li et al., 2017)	This thesis: See methods section
Epochs	100	100	100	80
Training sample	0.6	0.6	0.6	0.6
Selected samples	20858	20858	20858	24969
Sampling mode	Disjoint	Disjoint	Disjoint	Random
Patch Size	1	1	5	-
Learning Rate (LR)	0.0001	0.01	0.01	0.001
Batch size	100	100	100	32
Computation time (real) [s]	5432.5	347.6	5318.4	880 (training), ~10s (prediction /classification)
Accuracy [%]	89.8	75.2	93.9	98.43

4.2 Apliki

The results for the Apliki data confirm that it is possible to predict a sensible classification map of a mine face scan with a DeepGeoMap model that was solely trained on geochemically validated laboratory hyperspectral image data. For the Apliki laboratory samples, the OA would presumably be slightly lower by using the ROIs of figure 37, but still well over 90%, which can be regarded as an accurate classification. In the case of the Apliki sample data it was more clear when the DeepGeoMap model showed overfitting compared to the Brazilian iron ore samples. The model was trained for 50 epochs to find that optimal validation accuracy and loss. This was reached at 42 epochs. Before the classifications and loss showed signs of underfitting, and after 50 epochs overfitting became an issue. Hence The issue with reflected light form the plastic containers is very evident if the negative of the masked sample classification is inspected (figure 37). Around the sample areas are often misclassified by the DeepGeoMap model. This gets very evident if the background area (without shadow-masking) is masked out of the classification result (figure 43 A), or vice versa the classification of the rocks from the background (figure 43 B).

The issue with the misclassified plastic containers, in which the samples were placed, could probably be solved by simply adding more background data for class 8 (background) and then let the model train for longer as the complexity of the plastic

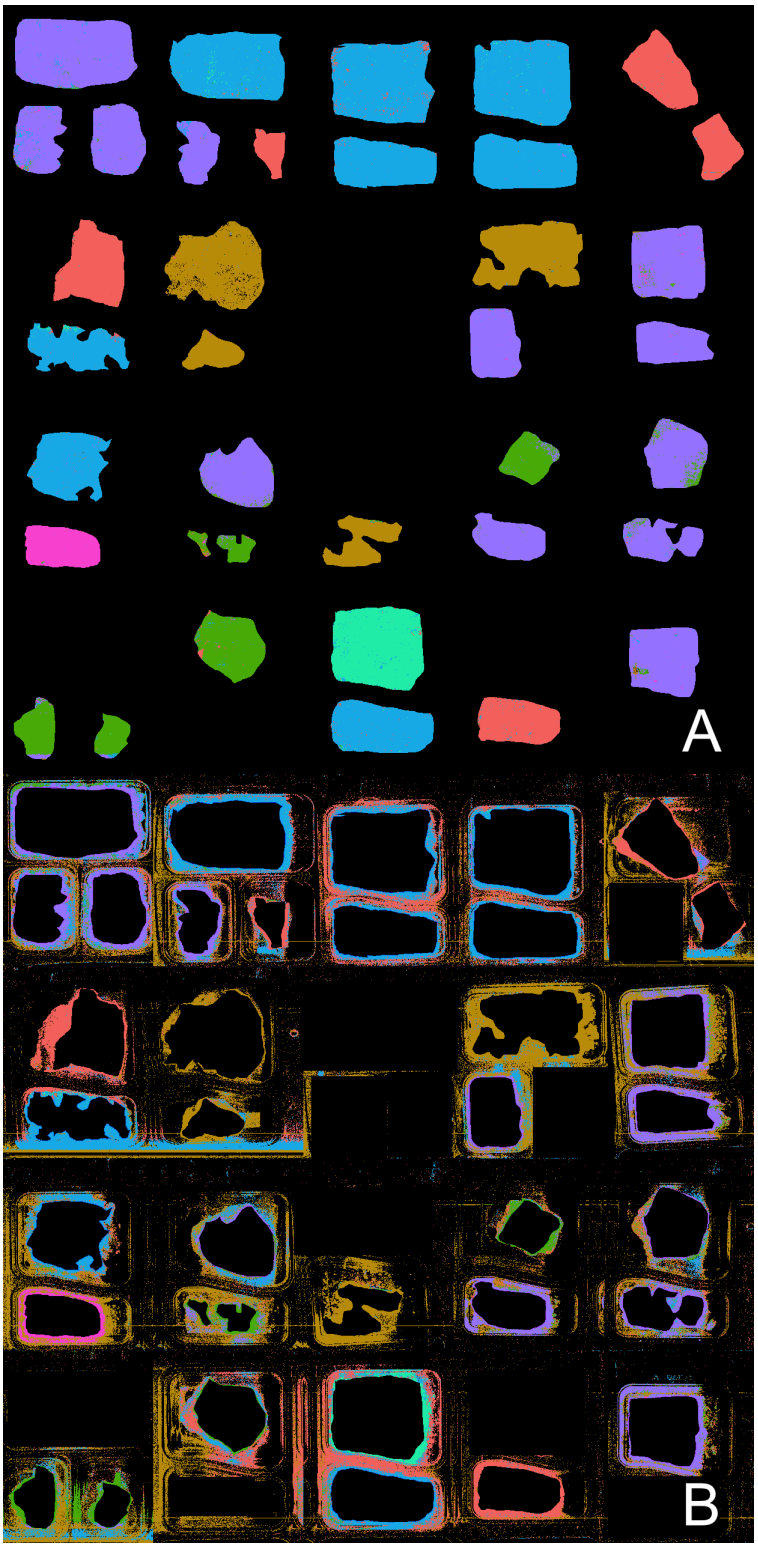

Figure 43: A: Apliki laboratory samples classification of an DeepGeoMap 42 epoch model, with an validation accuracy of 98.83%, where the areas outside of the ROIs shown in figure 37 B (the background without samples) has been masked out. The validation accuracy is not including the background classification (B); B: The negative of A, the classification image for the background areas with the areas of the sample validation masked out. Note that for a perfect background classification this image would have to be entirely black, with the exception of the very close direct surroundings of the rocks samples as the validation does not cover them completely (cross check: figure 37 A). Both images (A & B) together would again result in the complete classification image shown in figure 37.

reflectance seems quite high since it represents a mix of the plastic surface coating and the reflected spectra from the samples, while being very dark and hence noisy. Some of the reflected spectra were also classified as the most proximal rock samples, which could imply that that the DeepGeoMap model is in these few cases capable of identifying the spectra even under very distorted reflective conditions (as DeepGeoMap does not consider any spectral-spatial relationships). The spectrum of the shadows that the DeepGeoMap model trained with are also very close to the spectra reflected by the plastic containers (see figure 44), this and the noise is probably the reason why there are so many misclassifications in the background areas. Hence it is likely that it would require some experimenting and effort to improve the classification of the background when no shadow-masking of the data is wished for. This although might only be relevant for very specific applications. Hence it can be said that the ability to map rock samples in the shadowed areas more accurate, can lead to overfitting of other dark materials that were not part of the training input and can be seen as a tradeoff. With a better selection of background (ROI) sample input this could eventually be solved, since DeepGeoMap models can also differentiate with very high accuracy between the shadowed areas of the rock samples.

The classification capability of the 42 epoch DeepGeoMap model is in following compared to a Binary Feature Fitting and a ENVI Spectral Angle Mapper (SAM) classification of the Apliki laboratory data (figure 45). These represent the best classification results of this data set from the classifications conducted by Koerting (2021). Figure 45 A shows the predicted classification of the DeepGeoMap model after 42 epochs using 400 bands, with an validation accuracy of 98.83%. Figure 45 B shows the shadow masked classification result of Binary Feature Fitting (BFF) using the 400 bands, and a 36 spectra library, with a classification accuracy of 81,39%, C shows a 400 band shadow masked classification of the ENVI Spectral Angle Mapper (SAM) with 36 input spectra achieving 85.13% accuracy, and D shows the shadow masked classification result of BFF which is utilising a 70 band spectrally downsampled version of the image, and 36 input spectra, with an accuracy of 85.6%. Images B, C, D include samples that were not part of the 36 geochemically verified samples, hence these were ignored and not part of the training input for the DeepGeoMap classification (see section 2.3.2. While this shows that DeepGeoMaps model can predict the rock sample classes more accurate it also shows that the background classification does not work as good. Looking exclusively on the rock sample and background classifications (shown in figures 46 and 47) this becomes more evident. The reason for these background errors by the DeepGeoMap model is discussed above and is likely do to spectral similarity between the training input of shadowed rock samples and the reflections of the plastic containers. That this is due to spectral similarity and not overfitting is also backed by the fact that the Binary Feature Fitting 400 band image classification also shows very

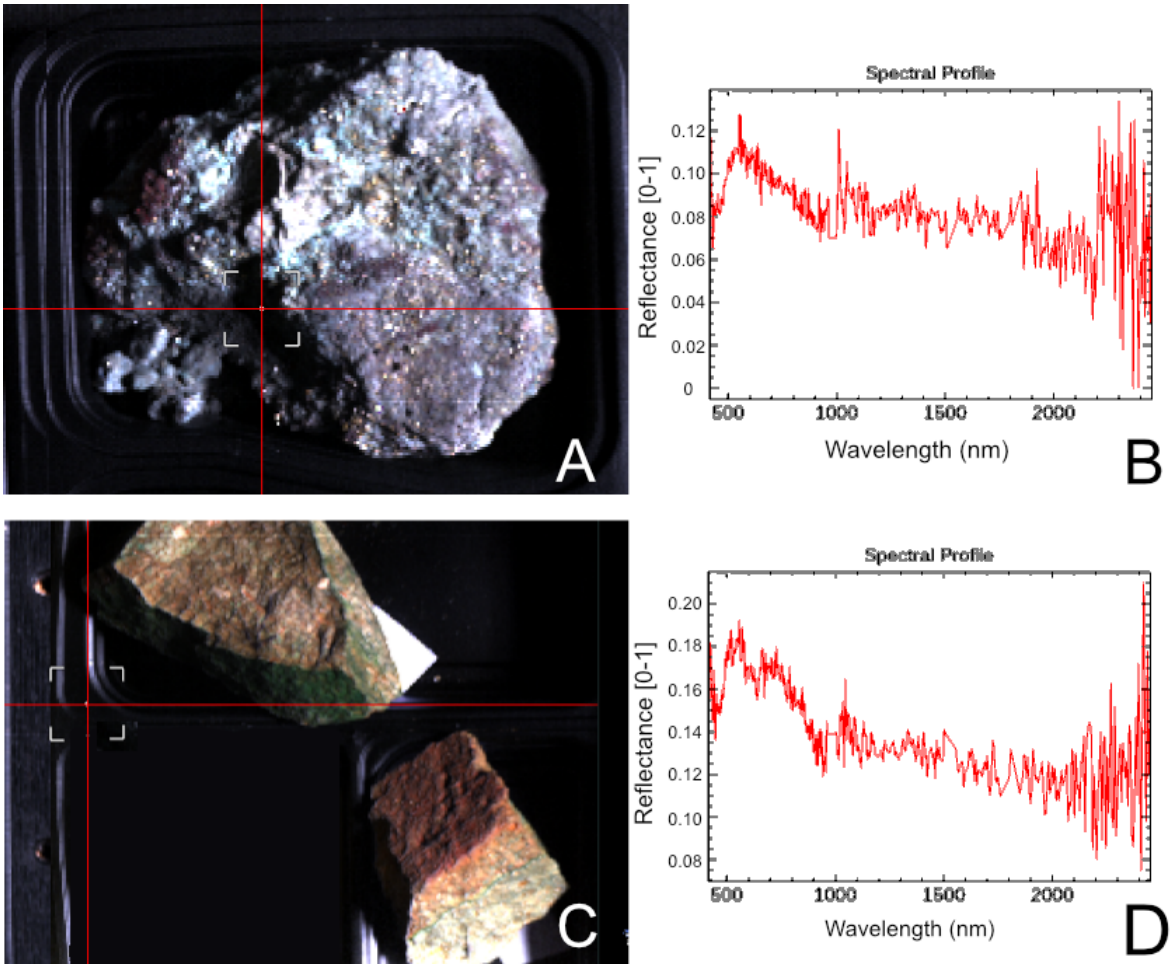

Figure 44: A: Indicating the location (red cross) of the spectral measurement shown in B for a shadowed part of a sample that has been input to the DeepGeoMaps model training. B: Spectra of a shadowed area that has been used for training. C: The location of the spectral measurement (red cross) shown in D for the edge of a plastic container. D: The spectrum of a plastic container edge, notice the noise, and the similarity to the spectra of the shadowed area of a mineral sample shown in B.

similar and often to the DeepGeoMap model matching background misclassifications (figure 47 B).

The results of the mine face scan are also shortly compared with two classifications conducted by Koerting (2021) and discussed (see section 3.2). This classification although is more open to geological interpretation and accuracies cannot be calculated so easily. The results of the 42 epoch DeepGeoMap model classification are shown next to the classifications of the Binary feature fitting (BFF) using 390 band and 36 library spectra and the Spectral Angle Mapper (SAM) classification using the same bands and spectral input. All three classifications correctly map the broader geological interpretation after Antivachis (2015) and Koerting (2021), with the exceptions that the Binary Feature Fitting (BFF) and the Spectral Angle Mapper (SAM) overestimate the occurrence of the gypsum mineralization likely due to atmospheric bands [24]. This

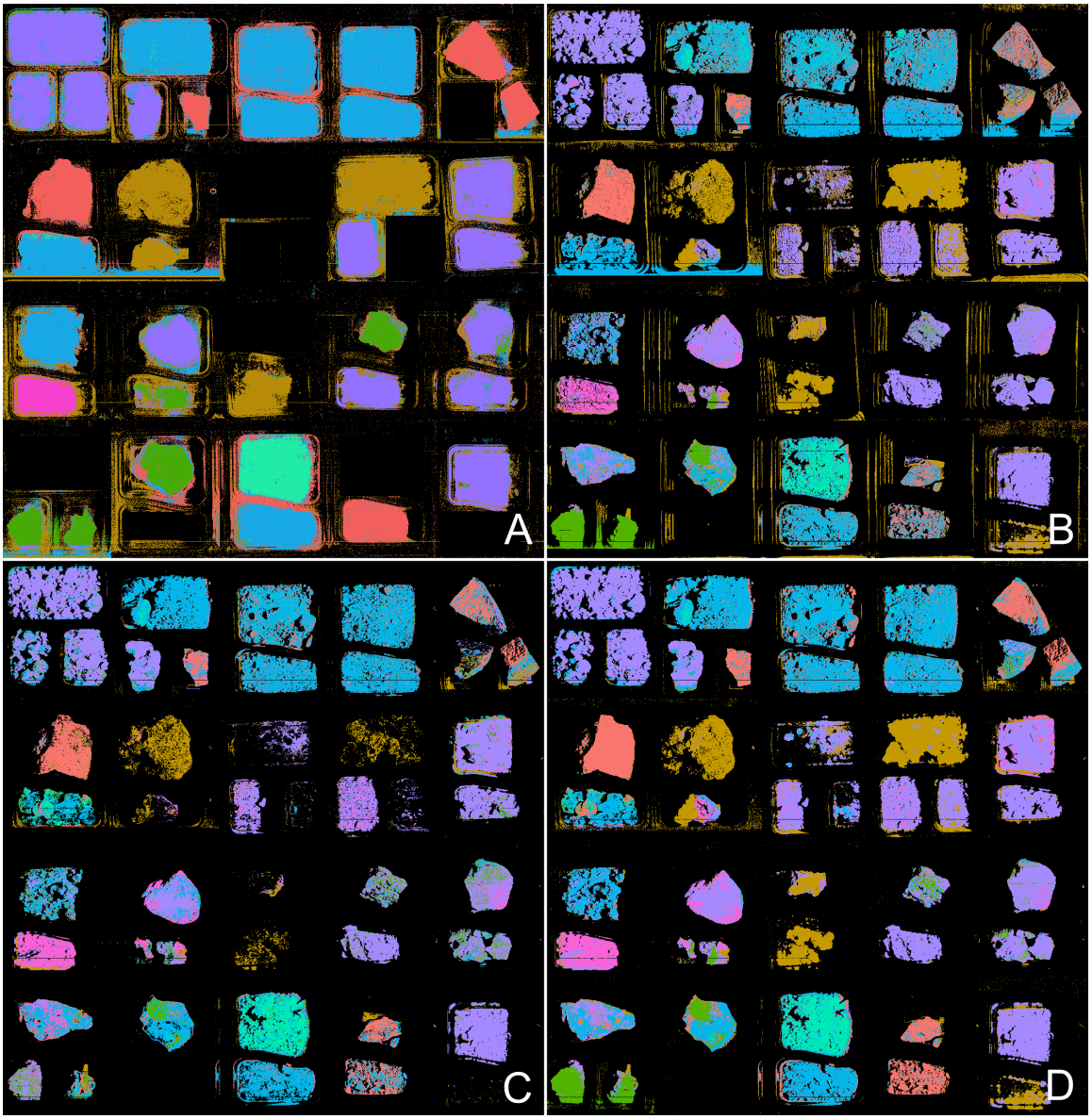

Figure 45: Overview of different classification methods for comparison of DeepGeoMaps classification capability for the Apliki Lab data. A: This image shows the predicted classification of the DeepGeoMap model after 42 epochs using 400 bands, with an validation accuracy of 98.83%; B: Shows the shadow masked classification result of Binary Feature Fitting (BFF) using the 400 bands, and a 36 spectra library, with a classification accuracy of 81,39%; C: shows a 400 band shadow masked classification of the ENVI Spectral Angle Mapper (SAM) with 36 input spectra achieving 85.13% accuracy. ; D: Shows the shadow masked classification result of BFF which is utilising a 70 band spectrally downsampled version of the image, and 36 input spectra, with an accuracy of 85.6%. Note that Images B, C, D include samples that were not part of the 36 geochemically verified samples, hence theiy were ignored for the DeepGeoMap classification. Images B, C, and D after Koerting (2021)

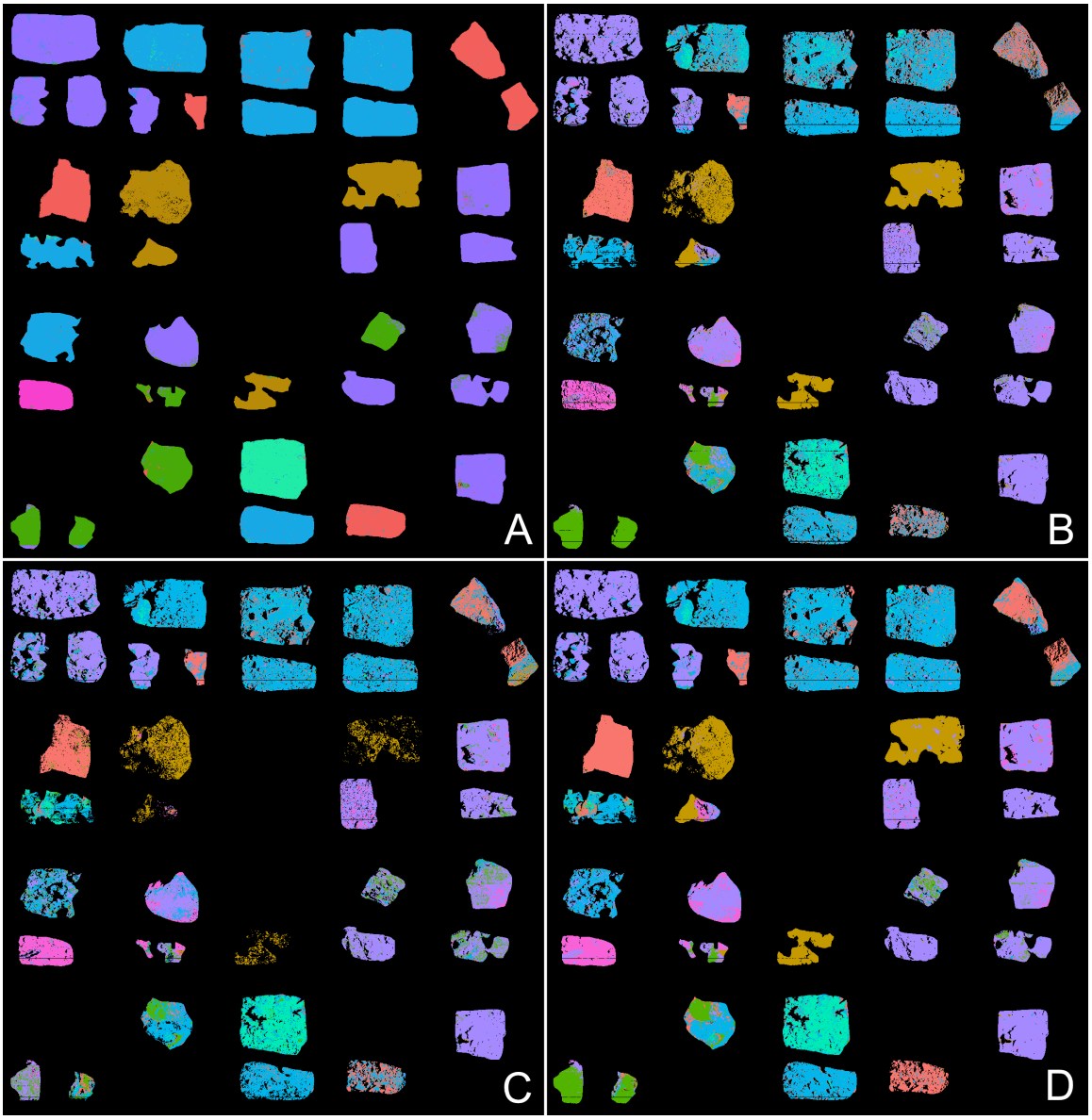

Figure 46: Overview of different classification methods for comparison of DeepGeoMaps classification capability for the Apliki Lab data. Same Images as in figure 45, except that the background of all images has been masked out. A: This image shows the predicted classification of the DeepGeoMap model after 42 epochs using 400 bands, with an validation accuracy of 98.83%; B: Shows the shadow masked classification result of Binary Feature Fitting (BFF) using the 400 bands, and a 36 spectra library, with a classification accuracy of 81,39%; C: shows a 400 band shadow masked classification of the ENVI Spectral Angle Mapper (SAM) with 36 input spectra achieving 85.13% accuracy. ; D: Shows the shadow masked classification result of BFF which is utilising a 70 band spectrally downsampled version of the image, and 36 input spectra, with an accuracy of 85.6%. Note that Images B, C, D have Images B, C, and D modified after Koerting (2021)

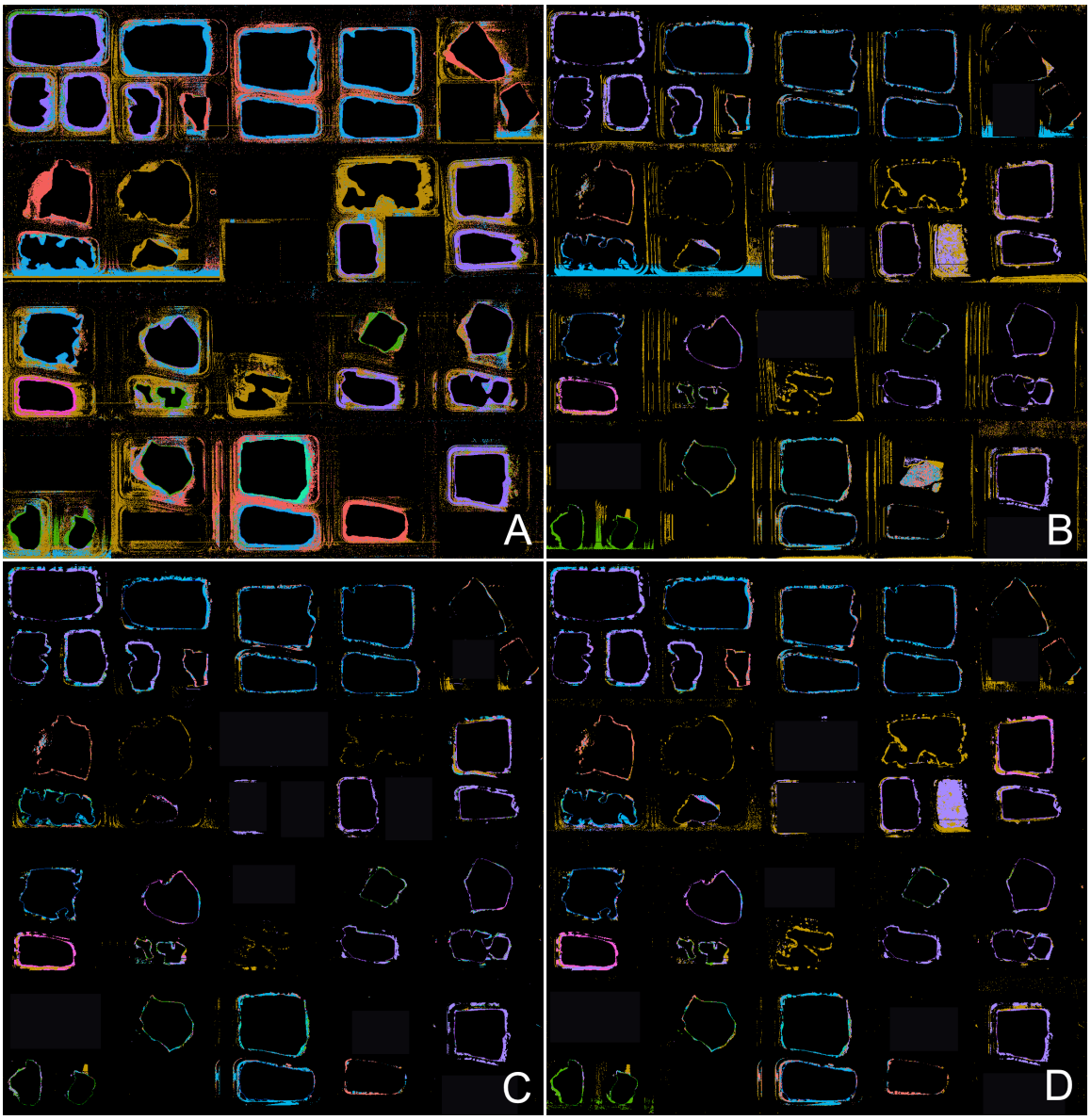

Figure 47: Error map of the background classification of different classification methods for comparison of DeepGeoMaps classification capability for the Apliki Lab data. Same Images as in figure 46, except that the rock sample validations of all images has been masked out. Note that for a prefect background classification this image would have to be entirely black, with the exception of the close direct surroundings of the rocks samples as the validation does not cover them completely (cross check: figure 37 A). Hence This shows roughly the misclassifications of the background of DeepGeoMap (A), BFF (B (400 bands) and D (70 bands)), and SAM (C). Images B, C, and D modified after Koerting (2021)

is probably not the case of for the DeepGeoMap classification as only smaller regions are mapped as gypsum. The DeepGeoMap model on the other hand, maps a lot of areas of cluster/class 3 in the middle of the open cut. This actually could represent the underlying geology as cluster 3 and cluster 6 both include chlorite-group minerals and which would be expected after the analysis of Koerting (2021) and Antivachis (2015). Overall it can be said that it is possible to create a geologicaly sensible mine face map from an hyperspectral image with a DeepGeoMap model that has been trained with geochemically validated laboratory hyperspectral sample data form that mine.

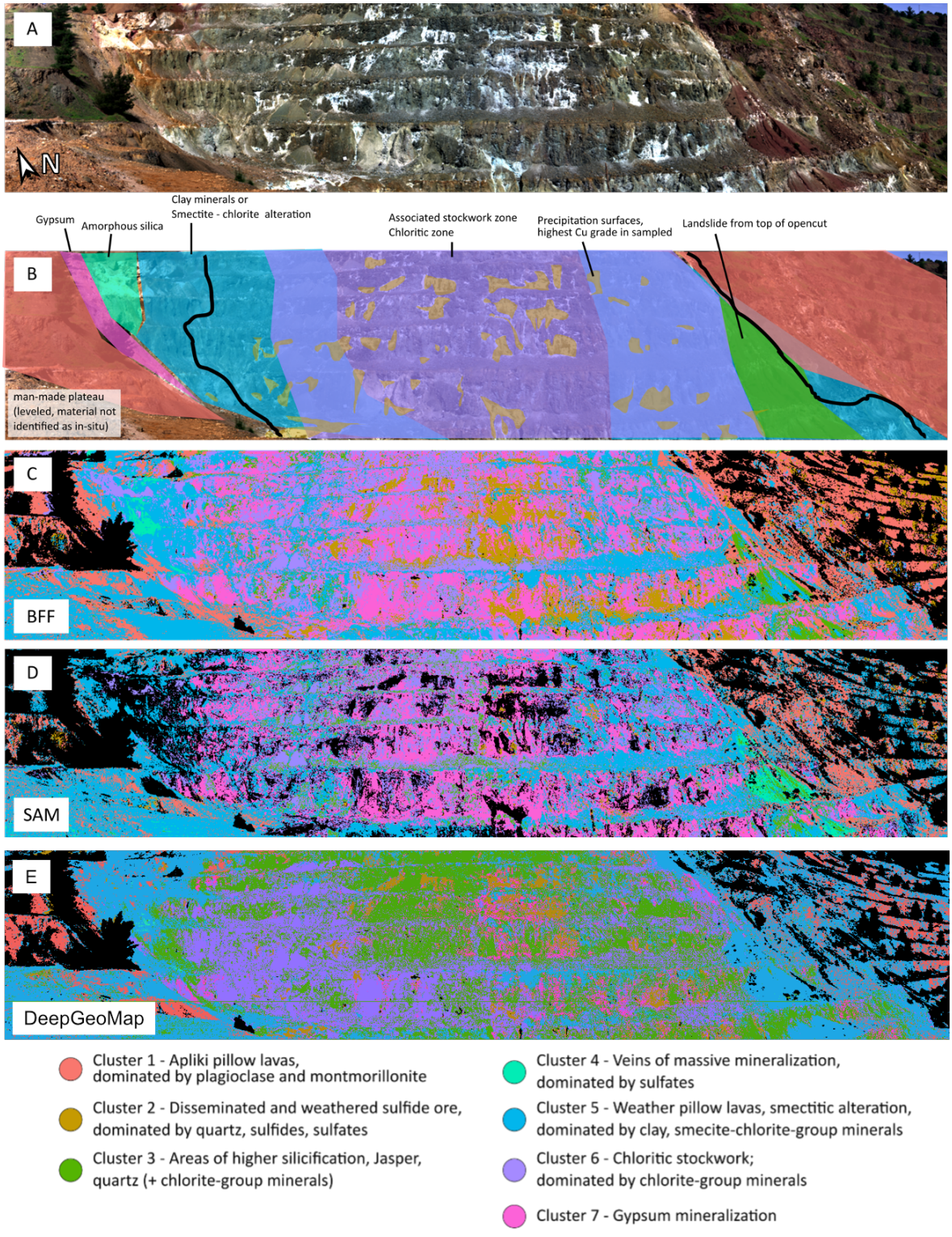

Figure 48: A: RGB representation of the Apliki open cut mine face, True color (R: 640nm – band 63, G: 549nm – band 38, B: 458nm – band 13); B: "the geological mapping of the open cut (B), based on Antivachis (2015)" C: Classification by of the Binary Feature Fitting (BFF) algorithm using 390 bands and 36 spectra. D: Classification of the Spectral Angle Mapper (SAM) using 390 bands and 36 spectra. Text, and the images A, B, C, and D after Koerting (2021). E: classification of the DeepGeoMap 42 epoch model using 390 bands.

4.3 Indian Pines Data Set

The Indian Pines data set is often used as a benchmark for hyperspectral image classification capabilities (see [41]). Since Indian Pines is a relatively old data set (from 1992) the spatial and spectral resolution is not very high compared to modern imaging spectrometer data. The more the classes vary spectrally and the less spectral variance occurs inside each class, the better 1D neural networks performs. Hence data with more difference between the classes and less difference within each class the better and more accurate the classification result.

The results from the DeepGeoMap model are compared to some of the most accurate neural networks spectral-spatial classifiers in regards to the Indian Pines data set. Note that the colormaps of classifications are adjusted to the publications DeepGeoMap is compared to.

Inspired by Sun et al. (2020) DeepGeoMap is compared with a Spectral Vector Machine (SVM) ([37]), the joint sparse representation classification (JSRC) ([11]), VGG ([48]), BASS Net ([46]), two-branch deep CNN (Two-CNN) ([55]), Dual-Tunnel CNN ([54]), and the localized spectral features and multi-scale spatial features convolution (LSMSC) by Sun et al. ([50]). Sun et al. (2020) explain the following: "SVM is a traditional pixel-wise classifier, whilst JSRC and all CNN-based approaches are spectral-spatial classification methods. BASS Net uses parallel CNNs to simultaneously learning spectral and spatial features from each spectral set, where the spectral features are obtained by band partitioning. It is worth noting that the Two-CNN and the Dual-tunnel CNN are similar to the proposed method, in which 1-D CNN and 2-D CNN are used to extract spectral and spatial features, respectively, and then fused in the full-connected layer." Sun et al. (2020) use a deep extraction of localized spectral features and multi-scale spatial features convolution (LSMSC) framework for spectral-spatial fusion based classification of hyperspectral images (HSIs) [50]. DeepGeoMap models (I), and all of the methods mentioned above (B-H) classifications of the Indian Pines validation data set are shown in figure 49. The accuracies are compared in table 9. Comparing the classifications and accuracies it becomes evident that DeepGeoMap is in the mid-range of these neural networks. While achieving much better classification result than the only other spectral-only classifier (SVM), the spectral-spatial methods work better for the Indian Pines validation data set.

The Indian Pines data set has frequently been used for benchmarking and most publications that show most accurate results for the Indian pines data set use 2D or 3D CNNs, or other techniques that incorporate spatial information of the data (see [41]. But it has to be noted that many publications benchmark the spectral-spatial classifiers often just on the the validation set not considering the actual scene. This is presumably because spectral-spatial classifiers take the spatial shapes into account.

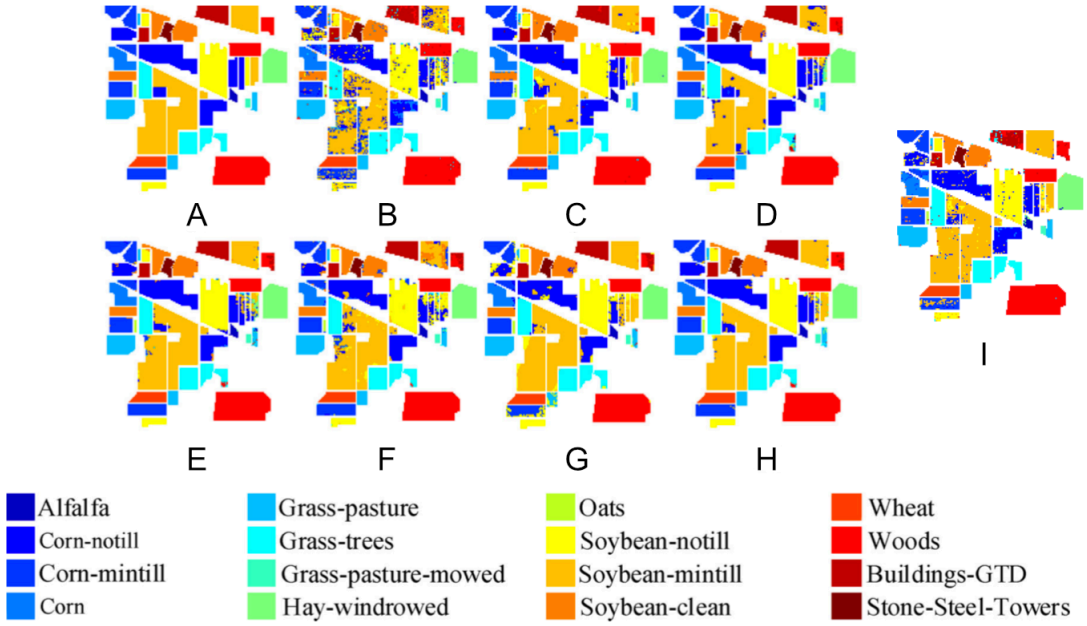

Figure 49: (A) Ground truth, (B) SVM, (C) JSRC, (D) VGG, (E) BASS, (F) Two-CNN, (G) Dual-tunnel CNN, (H) LSMSC, (I) DeepGeoMap. Images A-H and Legend after Sun et al. (2020)

Table 9: Comparison of the accuracies of DeepGeoMap and other methods shown in figure 49, modified after Sun et al. (2020).

	SVM	JSRC	VGG	BASS
OA	90.5	95.92	96.69	96.15
AA	96.23	97.22	98.11	98.23

	Two-CNN	dual-tunnel CNN	LSMSC	DeepGeoMap
OA	96.21	96.42	98.70	**96.21**
AA	97.89	98.36	99.35	**98.43**

While this works very well on the validation image (and hence also the regions of the validation image in the actual scene) the class and structural boarders of the actual scene classification often look a little dissolved and blur out small details of the full scene for example the very accurate deep&dense CNN by Paoletti et al.(2018), (see figure 50). The author of this thesis although notes that it is not possible by the available data of the Indian Pines set to determine statistically what is actually more correct for the classification of the full scene. The spectral-spatial classifiers show impressive results for the Indian Pines data set that are more accurate than the DeepGeoMap classifications.

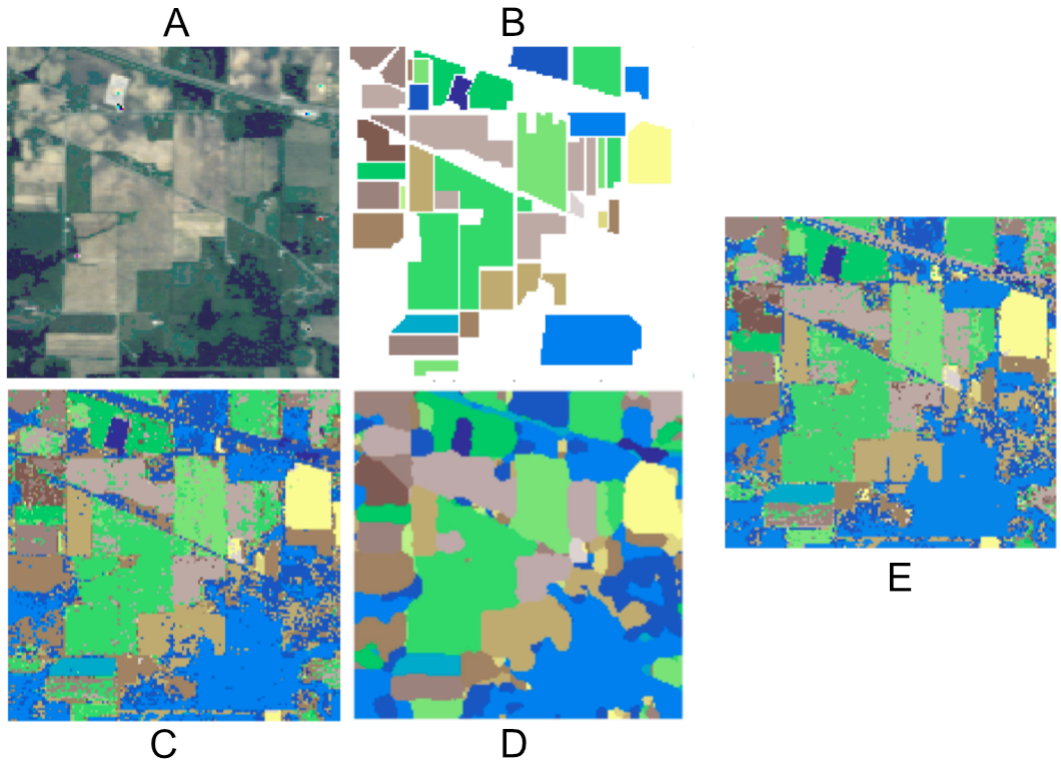

Figure 50: (A) RGB, (B) Ground Truth, (C) SVM (OA: 86.24%) [37], (D) deep&dense CNN (OA: 99.46%) [40], (E) DeepGeoMap (OA: 96.21%), Images A-D after Paoletti et al.(2018)

The data set also raises the question if there is any more information spectrally in the data that can be utilised by 1D neural networks. The neural network setup proposed for DeepGeoMap in this thesis is solely focused on the spectral information available in the data. This of course presents a disadvantage on classifying data that has limited spectral differentiability and additional spatial information like the Indian pines data set. DeepGeoMap as a one dimensional neural network focuses on purpose solely on the spectral information (see section 2.1) and hence was not expected to achieve the same (very high) accuracies as most modern spectral-spatial networks on this dataset as it shows little spectral variability across its classes and classification benefits highly from including spatial information such as shapes of the fields or structures. Also there are no ground truth spectra available that were recorded in

a laboratory. This means that DeepGeoMap cannot use its biggest advantages over differently structured networks, as they lie in its models being able to be trained with validated laboratory ground truth data which makes it possible to supply plenty of usually sparse ground truth training data to the DeepGeoMap models. Nevertheless it was still included in this work to show that DeepGeoMap can deliver acceptable accuracies (>90% OA) for difficult datasets just though spectral classification. Although not directly comparable due to a different number of training labels, and number of training pixels, it is demonstrated that DeepGeoMap shows roughly similar classification capabilities as other neural networks.

5 Conclusion

In this work it is shown that TensorFlow and Python was used to develop a deep convolutional neural network architecture that is able to classify hyperspectral images in a similar manner as classical algorithms for example like the EnGeoMAP, Spectral Angle Mapper, or the Binary Feature Fitting algorithm (see [38, 24]). It could also be shown that the deep neural network developed for this Masterthesis (DeepGeoMap) could achieve generally good classification accuracies for this type of network, surpassing the accuracies of classical algorithms and some other neural networks for some specific geological data sets presented here. The high amount of training data required in comparison to classical algorithms like EnGeoMap or Binary Feature Fitting (e.g. more than 20.000 spectra vs. 15 library spectra) can also present a disadvantage for data with very little training spectra available. The possibility to train DeepGeoMap models with large amounts of geochemically validated sample data and hence unmixed (or knowledgeably mixed) pixels and then use this laboratory data trained models to classify larger scale mine face, arial, or satellite imagery rapidly is seen by the author of this thesis as its biggest advantage over 2D or 3D convolutional neural networks, that also require large amounts of training data. To not include spectral-spatial relationships and shape information of the analysed materials could and can also prove a disadvantage. When there is a high amount of predictable spatial shapes in an hyperspectral image, DeepGeoMap clearly is in disadvantage over spectral-spatial relationship classifiers and/or classifiers that include predicting spatial shapes of structures, fields, or general morphology in the image data. It is shown that this although can change when the data presented (e.g. Brazilian iron ore) has much more unpredictable varying shapes (e.g. rocks), no characteristic larger spatial shapes (all samples were arranged in the same 4x4cm squares), and/or surface features (e.g. grain size) thats are not spatially differentiable at that resolution. Also when images of scenes are acquired at different distances and spatial-resolutions and the model training was only conducted on one of them, it cloud prove more challenging for spectral-spatial classifiers

than purely spectral classifiers such as DeepGeoMap, to map them with high accuracies. For example a DeepGeoMap model could be trained with beforehand acquired ground truth point-spectrometer data and/or laboratory image sample data (of rocks, powders, soil samples etc.) and can then use this model to rapidly classify any kind of hyperspectral target image that has the same number of spectral bands including mine face scans, mine conveyer belt images, satellite, arial, and drone images to name the most common. This spatially independent classification also makes position-dependent ground truthing less important as the collected samples or point spectrometer data just have to be in larger target region of e.g. an arial or satellite image, very similar to spectral library based classifications of classical algorithms such as EnGeoMap. DeepGeoMaps model training also can include a multitude of the spectra that have for e.g. been used in this thesis to train a model (if a high training time is acceptable), and still have relatively fast prediction/classification times, as model training and prediction/classification can be computationally independent as saving/storing the model is possible. It has to be seen if a 1D deep convolutional neural network approach like DeepGeoMap can utilise its advantage of point spectrometer and laboratory data training in one or some fields like scientific research, mineral resource exploration, mine conveyer belt scanning, etc. against 3D convolutional neural network architectures or others that include spectral-spatial relationships and of course shape information of analysed hyperspectral remote and proximal sensing data. For the hyperspectral geological image data presented here it was shown that DeepGeoMap is an accurate and reliable classification tool.

6 Outlook

Additional research is required in order to evaluate how well DeepGeoMap can actually utilize for example point spectrometer data for image classification and/or use its 1D principle and laboratory training to an advantage in more research campaigns. Especially interesting seems to further test the ability of DeepGeoMap regarding classification of satellite and drone image data with models that used laboratory samples for training. Testing DeepGeoMaps capabilities and limits with smaller and larger training set sizes seems also very interesting. Using larger data sets and more training input samples could create very accurate models that have 'seen' large amounts of sample sets, and after the time intensive training is completed are accurate and fast classifiers. A scenario where very fast and accurate ore type classification is needed and a model could be trained with very large amounts of hyperspectral laboratory data is a (mining or foundry) conveyer belt. General improvements could be made to DeepGeoMaps architecture as this is just a first version presented here, for example experimenting with convolutional kernel sizes to further suppress noise in the data or

further trying to improving prediction speeds. Finally, creating a user-friendly interface, where it is possible to input hyperspectral data, select training data with, and receive classifications from, seems like an interesting outlook.

References

[1] Acquarelli, Jacopo et al. "Spectral-spatial classification of hyperspectral images: three tricks and a new supervised learning setting". In: (2017). DOI: 10.3390/rs10071156. eprint: arXiv:1711.05512.

[2] Adamides, N. G. "Mafic-dominated volcanogenic sulphide deposits in the Troodos ophiolite, Cyprus Part 1 – The deposits of the Solea graben". In: *Applied Earth Science* 119.2 (June 2010), pp. 65–77. DOI: 10.1179/1743275811y.0000000001. URL: https://doi.org/10.1179/1743275811y.0000000001.

[3] Antivachis, Danae N. et al. "Secondary Sulphate Minerals in a Cyprus-Type Ore Deposit, Apliki, Cyprus: Mineralogy and Its Implications Regarding the Chemistry of Pit Lake Waters". In: *Mine Water and the Environment* 36.2 (Apr. 2016), pp. 226–238. DOI: 10.1007/s10230-016-0398-0. URL: https://doi.org/10.1007/s10230-016-0398-0.

[4] Antivachis, N. D. "The geology of the northern part of the apliki Cyprus-type ore deposit". In: *Bulletin of the Geological Society of Greece* 49 (Jan. 2017), p. 4. DOI: 10.12681/bgsg.11047. URL: https://doi.org/10.12681/bgsg.11047.

[5] Audebert, Nicolas, Saux, Bertrand, and Lefèvre, Sébastien. "Deep Learning for Classification of Hyperspectral Data: A Comparative Review". In: (Jan. 2019).

[6] Baumgardner, Marion, Biehl, Larry, and Landgrebe, David. *220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3*. en. 2015. DOI: 10.4231/R7RX991C. URL: https://purr.purdue.edu/publications/1947/1.

[7] Belkin, Mikhail et al. *Reconciling modern machine learning practice and the bias-variance trade-off*. 2018. eprint: arXiv:1812.11118.

[8] Carlos, Dionísio U., Uieda, Leonardo, and Barbosa, Valeria C.F. "Imaging iron ore from the Quadrilátero Ferrífero (Brazil) using geophysical inversion and drill hole data". In: *Ore Geology Reviews* 61 (Sept. 2014), pp. 268–285. DOI: 10.1016/j.oregeorev.2014.02.011. URL: https://doi.org/10.1016/j.oregeorev.2014.02.011.

[9] Carrara, S. et al. *Raw materials demand for wind and solar PV technologies in the transition towards a decarbonised energy system.* Publication Office of the European Union, 2020. DOI: 10.2760/160859. URL: https://data.europa.eu/doi/10.2760/160859.

[10] Chauvet, A. "Structural Control of Ore Deposits: The Role of Pre-Existing Structures on the Formation of Mineralised Vein Systems". In: *Minerals* 9.1 (Jan. 2019), p. 56. DOI: 10.3390/min9010056. URL: https://doi.org/10.3390/min9010056.

[11] Chen, Chen, Chen, Na, and Peng, Jiangtao. "Nearest Regularized Joint Sparse Representation for Hyperspectral Image Classification". In: *IEEE Geoscience and Remote Sensing Letters* 13.3 (2016), pp. 424–428. DOI: 10.1109/LGRS.2016.2517095.

[12] Dorr, John Van Nostrand. "Supergene iron ores of Minas Gerais, Brazil". In: *Economic Geology* 59.7 (Nov. 1964), pp. 1203–1240. DOI: 10.2113/gsecongeo.59.7.1203. URL: https://doi.org/10.2113/gsecongeo.59.7.1203.

[13] Education, IBM Cloud. *What is Overfitting?* 2021. URL: https://www.ibm.com/cloud/learn/overfitting.

[14] Eismann, Michael T. *Hyperspectral Remote Sensing.* SPIE, Apr. 2012. DOI: 10.1117/3.899758. URL: https://doi.org/10.1117/3.899758.

[15] Elshkaki, Ayman et al. "Resource Demand Scenarios for the Major Metals". In: *Environmental Science & Technology* 52.5 (Jan. 2018), pp. 2491–2497. DOI: 10.1021/acs.est.7b05154. URL: https://doi.org/10.1021/acs.est.7b05154.

[16] Geospatial, L3Harris. *ENVI.* 2021. URL: https://www.l3harrisgeospatial.com/Software-Technology/ENVI.

[17] Gómez, Raúl. *Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names.* May 2018. URL: https://gombru.github.io/2018/05/23/cross_entropy_loss/.

[18] Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. *Deep Learning.* http://www.deeplearningbook.org. MIT Press, 2016.

[19] Hinton, Geoffrey E. et al. *Improving neural networks by preventing co-adaptation of feature detectors.* 2012. eprint: arXiv:1207.0580.

[20] Huang, Gao et al. *Densely Connected Convolutional Networks.* 2016. eprint: arXiv:1608.06993.

[21] Kingma, Diederik P. and Ba, Jimmy. *Adam: A Method for Stochastic Optimization.* 2014. eprint: arXiv:1412.6980.

[22] Kiranyaz, Serkan et al. *1D Convolutional Neural Networks and Applications: A Survey*. 2019. eprint: arXiv:1905.03554.

[23] Kiranyaz, Serkan et al. "Progressive Operational Perceptrons". In: *Neurocomputing* 224 (Feb. 2017), pp. 142–154. DOI: 10.1016/j.neucom.2016.10.044. URL: https://doi.org/10.1016/j.neucom.2016.10.044.

[24] Koerting, Frederike. "Hybrid imaging spectroscopy approaches for open pit mining – Applications for virtual mine face geology". PhD thesis. University of Potsdam, 2021.

[25] Koerting, Friederike et al. *Hyperspectral imaging data of the northern mine face and of laboratory samples of the copper-gold-pyrite mine Apliki, Nicosia District, Republic of Cyprus*. 2021. DOI: 10.5880/GFZ.1.4.2021.001. URL: https://dataservices.gfz-potsdam.de/panmetaworks/showshort.php?id=075c9763-6d57-11eb-9603-497c92695674.

[26] Koerting, Friederike et al. *Mineral spectra and chemistry of 37 copper bearing surface samples from Apliki copper-gold-pyrite mine in the Republic of Cyprus*. en. 2019. DOI: 10.5880/GFZ.1.4.2019.005. URL: https://dataservices.gfz-potsdam.de/panmetaworks/showshort.php?id=escidoc:4573901.

[27] Kokaly, Raymond F. et al. "Mineral information at micron to kilometer scales: Laboratory, field, and remote sensing imaging spectrometer data from the orange hill porphyry copper deposit, Alaska, USA". In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (2016). DOI: 10.1109/igarss.2016.7730411.

[28] Kriesel, David. *Ein kleiner Überblick über Neuronale Netze*. 2007. URL: http://www.dkriesel.com.

[29] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (May 2017), pp. 84–90. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

[30] Laban, Noureldin et al. "Reduced 3-D Deep Learning Framework for Hyperspectral Image Classification". In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, Mar. 2019, pp. 13–22. DOI: 10.1007/978-3-030-14118-9_2. URL: https://doi.org/10.1007/978-3-030-14118-9_2.

[31] Lecun, Y. et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791. URL: https://doi.org/10.1109/5.726791.

[32] Li, Chenming et al. “Hybrid Dilated Convolution with Multi-Scale Residual Fusion Network for Hyperspectral Image Classification”. In: *Micromachines* 12.5 (May 2021), p. 545. DOI: 10.3390/mi12050545. URL: https://doi.org/10.3390/mi12050545.

[33] Lin, Min, Chen, Qiang, and Yan, Shuicheng. *Network In Network*. 2014. eprint: arXiv:1312.4400v3.

[34] Lin, Wan-Ju et al. “Evaluation of Deep Learning Neural Networks for Surface Roughness Prediction Using Vibration Signal Analysis”. In: *Applied Sciences* 9.7 (Apr. 2019), p. 1462. DOI: 10.3390/app9071462. URL: https://doi.org/10.3390/app9071462.

[35] Martin, Andrew J. et al. “Extreme enrichment of selenium in the Apliki Cyprus-type VMS deposit, Troodos, Cyprus”. In: *Mineralogical Magazine* 82.3 (Apr. 2018), pp. 697–724. DOI: 10.1180/mgm.2018.81. URL: https://doi.org/10.1180/mgm.2018.81.

[36] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[37] Melgani, F. and Bruzzone, L. “Classification of hyperspectral remote sensing images with support vector machines”. In: *IEEE Transactions on Geoscience and Remote Sensing* 42.8 (2004), pp. 1778–1790. DOI: 10.1109/TGRS.2004.831865.

[38] Mielke, Christian. “Multi- and Hyperspectral Spaceborne Remote Sensing for Mine Waste and Mineral Deposit Characterization, new Applications to the EnMAP and Sentinel-2 Missions”. PhD thesis. University of Potsdam, 2015. URL: https://gfzpublic.gfz-potsdam.de/pubman/item/item_2016907.

[39] Mielke, Christian et al. “EnGeoMAP 2.0—Automated Hyperspectral Mineral Identification for the German EnMAP Space Mission”. In: *Remote Sensing* 8.2 (Feb. 2016), p. 127. DOI: 10.3390/rs8020127. URL: https://doi.org/10.3390/rs8020127.

[40] Paoletti, Mercedes et al. “Deep&Dense Convolutional Neural Network for Hyperspectral Image Classification”. In: *Remote Sensing* 10.9 (Sept. 2018), p. 1454. DOI: 10.3390/rs10091454. URL: https://doi.org/10.3390/rs10091454.

[41] *Papers with Code - Indian Pines Benchmark (Hyperspectral Image Classification)*. URL: https://paperswithcode.com/sota/hyperspectral-image-classification-on-indian.

[42] Rencz, Andrew N. and Clark, Roger N. “Spectroscopy of rocks and minerals and principles of spectroscopy”. In: *Manual of remote sensing, Remote sensing for the earth sciences*. 3rd ed. Vol. 3. Wiley, 1999, pp. 3–58.

[43] Rockwell, Barnaby W. *Comparative mineral mapping in the Colorado Mineral Belt using AVIRIS and ASTER remote sensing data*. 2013. DOI: 10.3133/sim3256. URL: https://doi.org/10.3133/sim3256.

[44] Rogass, Christian et al. "Translational Imaging Spectroscopy for Proximal Sensing". In: *Sensors* 17.8 (2017). ISSN: 1424-8220. DOI: 10.3390/s17081857. URL: https://www.mdpi.com/1424-8220/17/8/1857.

[45] Ruder, Sebastian. *An overview of gradient descent optimization algorithms*. 2016. eprint: arXiv:1609.04747.

[46] Santara, Anirban et al. "BASS Net: Band-Adaptive Spectral-Spatial Feature Learning Neural Network for Hyperspectral Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.9 (2017), pp. 5293–5301. DOI: 10.1109/TGRS.2017.2705073.

[47] Scherer, Dominik, Müller, Andreas, and Behnke, Sven. "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition". In: *Artificial Neural Networks – ICANN 2010*. Ed. by Diamantaras, Konstantinos, Duch, Wlodek, and Iliadis, Lazaros S. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101.

[48] Simonyan, Karen and Zisserman, Andrew. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. eprint: arXiv:1409.1556.

[49] Srinivasamurthy, Ravisutha Sakrepatna. ""Understanding 1D Convolutional Neural Networks Using Multiclass Time-Varying Signals"". In: *All Theses.* 2911. (2018). URL: https://tigerprints.clemson.edu/all_theses/2911.

[50] Sun, Genyun et al. "Deep Fusion of Localized Spectral Features and Multi-scale Spatial Features for Effective Classification of Hyperspectral Images". In: *International Journal of Applied Earth Observation and Geoinformation* 91 (Sept. 2020), p. 102157. DOI: 10.1016/j.jag.2020.102157. URL: https://doi.org/10.1016/j.jag.2020.102157.

[51] Sun, Le et al. "Supervised Spectral–Spatial Hyperspectral Image Classification With Weighted Markov Random Fields". In: *IEEE Transactions on Geoscience and Remote Sensing* 53.3 (Mar. 2015), pp. 1490–1503. DOI: 10.1109/tgrs.2014.2344442. URL: https://doi.org/10.1109/tgrs.2014.2344442.

[52] USGS. *integrated hyperspectral, geophysical and geochemical studies of yellowstone national park hydrothermal systems*. URL: https://www.usgs.gov/centers/gggsc/science/integrated-hyperspectral-geophysical-and-geochemical-studies-yellowstone?qt-science_center_objects=0#qt-science_center_objects.

[53] Wan, Yu-qing, Fan, Yu-hai, and Jin, Mou-shun. “Application of hyperspectral remote sensing for supplementary investigation of polymetallic deposits in Huaniushan ore region, northwestern China”. In: *Scientific Reports* 11.1 (Jan. 2021). DOI: 10.1038/s41598-020-79864-0. URL: https://doi.org/10.1038/s41598-020-79864-0.

[54] Xu, Xiaodong et al. “Multisource Remote Sensing Data Classification Based on Convolutional Neural Network”. In: *IEEE Transactions on Geoscience and Remote Sensing* 56.2 (2018), pp. 937–949. DOI: 10.1109/TGRS.2017.2756851.

[55] Yang, Jingxiang, Zhao, Yong-Qiang, and Chan, Jonathan Cheung-Wai. “Learning and Transferring Deep Joint Spectral–Spatial Features for Hyperspectral Classification”. In: *IEEE Transactions on Geoscience and Remote Sensing* 55.8 (2017), pp. 4729–4742. DOI: 10.1109/TGRS.2017.2698503.

Acknowledgements

First and foremost I would like to thank my supervisors and mentors Dr. Christian Mielke and Prof. Uwe Altenberger for their kind support. I would like to express my thanks to all of the people who contributed to the successful completion of my master thesis at the University of Potsdam and GFZ Potsdam. My special thanks goes to Friederike Koerting for her advice, support and her permission to use her dissertation data sets to develop and test my neural network. My special thanks for the sample data used goes as well to Bertram Heinze and Geological Survey Department of the Republic of Cyprus.