Developing and Evaluating Simplified Tools for Image Processing in a Problem-Based Learning Environment for Earth Observation

Henryk Hodam¹ · Andreas Rienow¹ · Carsten Juergens¹

Received: 23 February 2022 / Accepted: 23 June 2022 / Published online: 14 July 2022
© The Author(s) 2022

Abstract
Earth observation is an interdisciplinary field of application. This makes it ideal for supporting natural science education in schools. Understanding satellite data can enable students to see more than just “beautiful images”. They can gain insights into an exciting field of application of many natural- and computer science fundamentals. In recent years, interactive lessons have been produced by our working group. They address the prospects and challenges connected with the teaching of earth observation topics in schools. The principles upon which the lessons are built promote a problem-based style of learning combined with a low impact in extraneous cognitive load for the students. Advances in web technology have made it necessary for interactive lessons to be redesigned and rethought. Contemporary web technologies have been selected and combined into an application framework to fulfill the requirements of our design principles and put a greater emphasis on usability during the content creation process. To allow offline usage of the learning modules it was tested whether this can be achieved using strictly client-side image processing. This article shows that, based on the technology stack used, lessons can be created that enable pupils to analyze remote sensing data in a much-simplified way. Using those simplified methods, a classification with an overall accuracy of 78.66% was achieved. The comparison of another simplified classification using just threshold values resulted in up to 89% probability to predict the outcome of a professionally produced dataset, making the simplified methods applicable in real-world examples presented in problem-based teaching scenarios.

Keywords Earth observation · Education · Schools · eLearning · Image processing · Learning environment · Digital teaching modules

Zusammenfassung
Entwicklung und Evaluierung von vereinfachten Werkzeugen für die Bildverarbeitung in einer problemorientierten Lernumgebung für Erdbeobachtung. Die Erdbeobachtung ist ein interdisziplinäres Anwendungsfeld. Damit eignet sie sich ideal zur Unterstützung der naturwissenschaftlichen Bildung in Schulen. Ein grundlegendes Verständnis von Satellitendaten kann es Schülerinnen und Schülern ermöglichen, mehr als nur „schöne Bilder“ in ihnen zu sehen. Vielmehr ermöglichen sie einen Einblick in ein spannendes Anwendungsfeld vieler natur- und informationswissenschaftlicher Grundlagen. Speziell für den Einsatz in Schulen haben wir in den letzten Jahren interaktive Unterrichtseinheiten entwickelt, die Möglichkeiten aufzeigen, die Arbeit mit Erdbeobachtungsthemen in der Schule einzubinden. Die Konzepte und Prinzipien, auf denen die Lernmodule aufgebaut sind, ermöglichen ein Lernen, das Problemorientierung mit einer geringen extrinsischen kognitiven Belastung für die Nutzer kombiniert. Fortschritte in der Webtechnologie haben eine Neugestaltung und Neukonzeption dieser interaktiven Unterrichtsmaterialien erforderlich gemacht. Aktuelle Technologien wurden ausgewählt und zu einem Framework kombiniert,
um den Anforderungen des Schulumfeldes gerecht zu werden und gleichzeitig die Benutzerfreundlichkeit während der Arbeit mit den Bildern zu erhöhen. Da die Module zukünftig auch als Offline-Lernmodule eingesetzt werden sollen, kam dabei eine rein clientseitige Berechnung der Bilddaten zum Einsatz. Dieser Artikel zeigt, dass auf Basis des verwendeten Technologie-Stacks erfolgreich Unterrichtseinheiten erstellen können, die es Schülern ermöglichen, Fernerkundungsdaten auf einfache Weise zu analysieren. Mit diesen vereinfachten Methoden zur Bildanalyse konnte eine Klassifizierung mit einer Gesamtgenauigkeit von 78,66% erreicht werden. Der Vergleich einer anderen vereinfachten Klassifizierung, die nur Schwellenwerte verwendet, ergab eine Vorhersagewahrscheinlichkeit von bis zu 89% für das Ergebnis eines professionell erstellten Datensatzes. Die Ergebnisse zeigen, dass die Werkzeuge mit professionellen Softwarelösungen und Methoden vergleichbar und somit auf reale Beispiele in problemorientierten Unterrichtsszenarien anwendbar sind.

1 Introduction

Image processing is the core of the analysis of remote sensing data. Besides being an important tool that is used by scientists and commercial enterprises alike to draw essential conclusions about the interaction of human activities with our environment, it also offers the chance to add a new perspective on natural and environmental sciences for pupils. These possibilities are emerging out of the interdisciplinary nature of remote sensing itself. While physics, maths, and computer sciences are from the methodological foundation of earth observation, their applications mainly stem from environmental topics connected to geographical or biological questions. This combination matches the fields taught in natural sciences in schools and therefore offers a range of opportunities to introduce pupils and teachers alike to the vast possibilities remote sensing is offering. To enable pupils to access this potential it is necessary to provide them with an environment offering easy access to data and drastically reducing the complexity of the methods and tools used to analyze the data. In this paper, we demonstrate that it is possible to achieve this on a technological level and produce products by simple means of image analysis that appropriately represent real-life scenarios in problem-based learning instruction.

2 Related Work

To understand how the results of this work can be related to similar approaches, it is necessary to take look at the options a teacher interested in using earth observation and image processing has and what other groups in the field of remote sensing conduct similar approaches. Classic image analysis in remote sensing is done in desktop software using various software solutions. The use of software in schools, however, is often connected to cost and administration efforts as well as aspects of teacher training. In combination, these barriers would prevent the use of image processing in classrooms straight away (Goetzke et al. 2013). The free Tools SNAP (SNAP) and LEOWorks (LEO Works) developed by ESA mitigate the cost factor and are usable on special occasions like project days but the necessity of training, data acquisition as well as the lack of available material make them cumbersome in everyday schools situations. Projects conducted in Italy and Poland presented successful ways to implement the use of software in extended periods of project work on the analysis of earth observation data in schools and used SNAP and additional software solutions (Amici and Tesar 2020; Dziob et al. 2020). Similar experiences were also made in our working group showing that under special circumstances and with professional assistance the use of software can be an option. The drawbacks of this approach are the high demand of personnel in relation to the pupils reached by the activity as well as the preparation time needed. A study on teacher motivation to use earth observation in their teaching of climate change showed that short and easy-to-use material is preferable (Asimakopoulou et al. 2021). The benefit of using software is its independence from a high-speed internet connection which still can be an issue today in German schools (forsa Politik- und Sozialforschung GmbH 2019). Recent developments are trending towards web-based solutions. WebGIS applications are available for almost two decades now and also systems focused on their use in schools were developed (Düren and Bartoscheck 2013). Early on the integration of remote sensing data into WebGIS was practiced (Tsou et al. 2005) and the availability of free satellite images on platforms like Google Earth demonstrated the public demand for satellite-based geographical information. But it was not until the advent of cloud computing methods that web-based solutions for remote sensing image processing became widespread (Gomes et al. 2020). Today doing advanced analysis in cloud-based environments like Earth Engine (Gorelick et al. 2017) is commonplace and online applications like the EO Browser (Sinergise Laboratory for Geographical Information Systems, Ltd. 2017) are making the instant analysis of datasets accessible to a larger audience, and also showing potential for the use in schools by offering an educational setting with reduced functionalities. There are, however, no materials available yet that would help teachers integrate this online tool into their teaching, and even with reduced functionalities, tools are abundant and cannot be reduced to fit certain topics. Approaches from Italy and Germany that try
new solutions to implemented teaching at graduate level and also emphasizes the importance to use real world scenarios in the teaching. (Gerner and Pause 2020; Maggioni et al. 2020). There are also approaches on digital teaching environment focused on rote sensing but they are also aimed at graduate level training (Eckardt et al. 2018; He et al. 2019).

The approach closest to the one presented in this work is the online tool BLIF (Ditter et al. 2010) which was designed with pupils and teachers in mind and offers datasets and tools that are tailored to predefined lectures, making it a valuable asset in integrating remote sensing in class and is used until today in different educational setups (Dannwolf et al. 2020). It is, however, solely focused on geography lessons. Another recent approach to integrating remote sensing into classrooms makes use of the BYOD (Bring Your Own Device) concept, integrating the capabilities of pupil’s smartphones in learning experiences based on augmented reality (Lindner et al. 2019, 2022; López Núñez et al. 2020; Rienow et al. 2020). While they show promising capabilities and offer interesting approaches to even include topics like hyperspectral imaging in classrooms (Lindner et al. 2022) they are limited in terms of computational capacities of the devices used.

3 Research Questions and Structure

3.1 Requirements

The approaches and tools described in Sect. 2 each show promising results for specific use cases, and often present learning material and modules focused on one topic. Few of them, however, offer a consistent environment to create and embed teaching materials and none offers potential for applications addressing all STEM subjects. Since this is the gap this work tries to fill the requirements that emerged out of some basic conditions dictated by the goal of using image processing as a part of STEM in schools is described hereafter.

While satellite images are used in German schools (Siegmund 2011) they are often not used as digital data but as simple images and mainly in geography classes. Therefore the potential they might have in education is not fully accessed (Goetzke et al. 2013). To solve this problem various interactive materials that try to mitigate several barriers of the school environment have been designed in the past to apply the potential of image analysis to the whole range of natural sciences by connecting topics of the curriculum to different aspects of remote sensing (Goetzke 2014; Rienow et al. 2013; Voß et al. 2013). They address topics found in the STEM curricula of all levels of German high schools and already played an important part in creating a school curriculum centered on remote sensing and are still in use in this context (Müller 2021). It has been shown that this can be achieved by tailoring the way tools and data are used by pupils and teachers to the special needs of the educational environment in schools that is limited by the time teachers can invest (Hardwig et al. 2018) as well as the technical equipment of the schools (Hodam et al. 2020). To be able to create lessons that are not only focused on the application side of remote sensing but also target fundamentals of physics, maths, or computer sciences, it is necessary to develop tools that depart from the usual functionalities of image analysis for remote sensing. Being, for example, able to read binary values from pixels normally hardly makes sense, but is a perfectly suitable tool for teaching the binary system in connection with image analysis in computer sciences. To unlock the potential satellite images may have for a modern approach to learning the modules are rooted in two didactical paradigms. First, they follow the didactical concept of problem-based learning. It is based on the principle of constructivism (Schunk 2019; Terhart 2003) and tries to stimulate knowledge construction by presenting a problem and letting the pupils figure out solutions largely on their own while solving the problem in an authentic if possible real-world scenario (Savery and Duffy 1995). While this open approach to knowledge construction seems to fit well for working with satellite data its open and undirected nature can also lead to frustration amongst learners (Mayer and Moreno 2003). So the second paradigm was controlling the amount of information as well as reducing the possible options for failure and thus lowering the extraneous cognitive load according to the cognitive load theory by Sweller (1988).

Learning environments created by our working group in the last decade incorporate these requirements but are as of now outdated. They are defined as “Integrated Learning Environments” (Hodam et al. 2020). Since the situation regarding the technical equipment and human factors in the digitalization process in German schools changed little in recent years and still leaves a lot to be desired (forsa Politik- und Sozialforschung GmbH 2019), the basic concept of how pupils can be brought in contact with remote sensing largely remains unchanged. There are, however, certain factors that made it necessary to rethink the technological aspects of the creation as well as the application of those materials. On the creation side, a framework is used that allows the authoring of the interactive teaching materials by combining data, tools, and background information into an interactive learning module. The framework is operated with Adobe AIR and Adobe Professional as development environments. Both assume a quite sophisticated level of training to be used efficiently. Without some knowledge of programming, content creators cannot work with the development environments which makes it hard to try and test new ideas and to focus more on the content in general. This drawback
makes a more efficient and easy-to-use solution necessary. The application side both profits and suffers from certain possibilities as well as limitations of the technology used. The learning environments can be run as a desktop as well as a web application. This is well suited to the application scenario since insufficient internet speed was and still is an issue in schools (forsa Politik- und Sozialforschung GmbH 2019) and the desktop applications run completely offline.

But since the modules, based on the flash web technology, are programmed in ActionScript 3.0 (Adobe) and result in applications in the proprietary Flash format that, since 2020, is not supported by web browsers anymore they can no longer be used online. Also, the technology does not allow for the use of genuine geospatial image data or web mapping methods, without an unreasonable development effort. Therefore, all image data have to be implemented as standard image formats such as JPEG or PNG (Randers-Pehrson 2003). It also lacks standard functionalities like panning or zooming and, above all, georeferencing that allows pupils to put the location of the image data in a global context. New developments in the field of web technology, however, opened up opportunities to create and implement functionalities that were not possible before and also are fitting to avert the limitations in both the creation and the application of the learning modules. Those opportunities, however, are connected to challenges. In development, the main difference to the technology used so far is that web applications no longer rely on just one main technology but rather a stack of methods, libraries, and programming languages as opposed to the almost single-branched technology tree that was used to develop the teaching materials before. And while this results in far more versatile applications it also adds to the level of complexity in development. Moreover, the stack of technology that meets the demands of the creation and application of the modules also has to be capable of producing image processing results comparable to real-world scenarios while reducing the complexity of the tools used. The work presented in this paper shows a way to achieve this while maintaining the validity of the data that can be produced, which can be then used to analyze real-world scenarios in problem-based learning units.

### 3.2 Research Question

The basic assumption to achieve the aforementioned goal on a technical level also represents the hypothesis of this paper:

**Browser-based tools for educational remote sensing image analysis with a reduced complexity in terms of tools and data can provide results that are sufficiently accurate to represent real-world application scenarios.**

To validate this hypothesis two research questions are formulated and answered in this work:

1. Does a flood hazard map produced with tools reduced in complexity generate a result comparable to an official flood hazard map?
2. Does a classification produced with tools reduced in complexity generate comparable results to professional software?

To answer the research questions regarding the accuracy of a flood hazard map produced with the tools embedded in the learning environment the following conditions should be met:

- The visual conclusion of a site analyses conducted in a problem-based learning scenario in the environment leads to the same conclusion as an openly accessible flood hazard map.
- In a statistical comparison of both flood hazard maps, the flooded and non-flooded areas in the hazard map derived from the tool predict the flooded and non-flooded areas of official data to a percentage of 80% and higher.

To answer the research questions regarding the classification produced with the tools embedded in the learning environment the following condition should be met:

- The overall accuracy of the classification tool matches the accuracy achieved with the same algorithm in a professional software.

It is also important to point out that the possibilities the tools can offer for the production of digital lessons for earth observation are the result of the technological framework they are embedded in. They thus depend on it as a methodical base upon which the research question can be answered. So to confirm if the technology stack is indeed able to provide the necessary functionalities it is used to create an exemplary interactive teaching unit that is also presented as proof of concept.

Section 4 describes the technologies used in the creation of the framework and links it back to the requirements that are described in Sect. 3. It also will present the methods and data used to assess whether real-world scenarios can be reproduced by the framework.

In Sect. 5, the results are presented in two parts. The first will document how the technologies introduced in Sect. 4 are combined into an application by presenting both the content management and the frontend application of an exemplary learning module. The second part will show the results of the comparisons performed on data produced by the application framework and software or official datasets to answer the research questions. In Sect. 6, the results are interpreted in regard to the research question and hypothesis.
and a conclusion based on the results is given in Sect. 7 alongside an outlook.

4 Data and Methods

Since the technology stack and the way it is combined to create this system sets the foundation for the capabilities and limitations of the teaching modules and thereby is the most important factor when it comes to the interpretation of the results, it is described in detail in this section. Also, the methods used to compare the results from the learning modules to those of professional software and official data are introduced in this section. The framework uses a variety of technologies, and the analysis of this technology stack is separated into two parts. The first describes the technology on the authoring side and the second focuses on the application side. To better separate the learning modules from the mechanisms of the content management they are referred to as the “Application” and the “CMS” (Content Management System).

4.1 Content Management

The system employed to create and manage the content of the learning applications has to meet several requirements:

1. It has to be modifiable and accommodate all content types and settings required to configure and populate the learning modules.
2. The content configuration has to be accessible from the outside via an Application Programming Interface (API).
3. It has to be compatible with certain common content types used in eLearning like H5P, an exchange format for rich html5 content, videos and HTML5 content in Iframes.

Developing a new system that would meet those requirements would have resulted in a disproportional effort so systems were examined that are already in use and could be adapted to our cause. A common approach to handling online learning content are LMS (Learning Management Systems) which provides the infrastructure for teachers to create, implement and manage their online content for their classes. There are numerous solutions available (Karadimas 2018) but while some are versatile concerning the implementation of digital materials none offers the level of modifiability that would allow the integration of complex functionalities that are needed for digital image analysis. Instead of integrating the actual learning modules into a system that would limit its capabilities, it was decided to split the learning application from the configuration of the learning modules using a CMS (Content Management System).

Based on the technical requirements there are several options to choose from. Commonly used CMS like Drupal (Buylaert 2022) or WordPress (WordPress Foundation 2022b) offer the abovementioned capabilities. Also, so-called “headless” CMS started to enter the market in recent years. Those completely omit a pre-defined frontend for content output for the sake of improved flexibility. At a first glance, those systems seemed well suited for the task, and first prototypes relied on the Wagtail headless CMS (Torchbox 2022). This approach, however, was abandoned due to limitations in support of requirement 3. In the end, WordPress was chosen because the way content is created is considered easier due to its widespread use. It also can be modified using JavaScript as the main language as opposed to i.e. Drupal which relies on PHP for this purpose, which helped the overall process because no additional programming language was introduced. Also, since it is very common, a lot of plug-ins already exist including the H5P (Joubel 2013) plugin that was needed for certain content types. The WordPress database can also be accessed via the WordPress-API allowing an alternative and versatile way to get information from within or outside the system to configure the learning modules.

4.2 Application

The content, as well as the functionalities that the main learning application offers, are set in the CMS as illustrated in Fig. 1. Just as the CMS, the application framework has to fulfill certain requirements that emerge out of the special demands based on the application scenario in schools.

1. The image data have to be georeferenced geospatial data in GeoTIFF format.
2. The image data have to be processed in the front end to enable future offline use if classroom bandwidth is limited.
3. The GUI (Graphical User Interface) has to be based on a modern framework for designing and programming user interfaces to simplify development and future modification.
4. The framework has to be extendible to be able to accommodate future scenarios and tools.

There is a wide selection of possible libraries and so-called “Maps Application Programming Interfaces” that can handle the display and interaction with geospatial data in web environments (Fernandes et al. 2013). The decision of which one to choose was mainly dictated by its ability to work with well raster data in general, its extendibility but also how well it works in conjunction with the GUI.
framework of or choice which is vue.js (You 2019) and will be described later in this chapter. A try and error approach helped us to decide for OpenLayers (Geospatial Foundation). Though most of the actual analysis is done by other libraries and the application framework uses mostly the basic mechanics of OpenLayers to display georeferenced data inside a layer logic. It also uses its ability to manipulate raster data for visually enhancing images.

A core requirement for the resulting learning modules is their ability to perform simple image analysis and thereby it depends on the use of genuine satellite image data. Displaying pre-calculated images would be an option for certain more instructive learning content but it would not be suitable for problem-based learning scenarios. To perform analysis on remote sensing images like, for example, calculating the Normalized Difference Vegetation Index (NDVI) it is necessary to use original uncompressed reflectance data stored in a georeferenced raster format. The GeoTIFF (Devys et al. 2019) format is widely used in software for raster data analysis and is considered to be the most popular format to exchange data between software. It is, however, not natively supported by browsers. For this reason and also due to performance issues as well as better access to large datasets, this kind of data is typically processed on the server-side and the web browser displays the outcome. Since our setup aims for the possibility to use the learning modules without the support of server-side calculations to make them potentially available as desktop applications this used to be an insurmountable barrier. Analyzing such rasters on the client-side is still an approach that is in its infancy because of the limitations of JavaScript being incapable of processing raw binary data until the introduction of the ECMAScript 2015 (ECMA International 2015). Also enabling the browser to use the hardware acceleration of the machine it runs on is a quite recent development. (Farkas 2020; Yao et al. 2018). A library to make those files accessible for web applications is geotiff.js (Schindler 2022). It allows parsing of TIFF files in general and thereby provides the basis for any in-depth image analysis on the client.

Image data have to be visualized after parsing. Two scenarios are covered in the system. The first is the displaying of the image data and the respective bands themselves and the second is the displaying of results that are stored as array data. Displaying anything in modern browsers relies on the canvas system of HTML5. The geotiff.js library includes a method to create an RGB composite canvas out of the raster data of a GeoTIFF file. The array of data resulting from calculations like the NDVI can be plotted as a canvas using the visualization library for scientific data plotty.js (Santillan 2022).

The processing of raster data at a reasonable speed in the browser is achieved using the JavaScript raster processing engine GeoBlaze (Dufour 2021). It extends the aforementioned geotiff.js library and enables the computation of basic raster statistics but also most importantly band arithmetic and thereby offering the possibility to perform analysis connected to indices and thresholds.

Besides addressing the needs of remote sensing and WebGIS functionalities, the framework includes a structure that contains and controls the learning process. The application itself is not part of the CMS. All information is stored in the database of the CMS and made available to the application through an API that compiles all information into a JSON (ECMA International 2017) file that is

---

Fig. 1 The data communication structure of the framework
interpreted as an object by the application. Everything is combined into a web application providing a GUI and a general structure to house both data and learning assets as shown in Fig. 1. The process of programming web applications today is dominated by the use of frontend web frameworks. Those frameworks open up new possibilities in the creation of complex dynamic websites and applications for a variety of purposes (Curie et al. 2019; Syafiq et al. 2018). In development, a framework like this offers the advantage of using components as the smallest parts of an application. These components can be instantiated as often as needed by the application and are updated dynamically (Song et al. 2019). Using those capabilities is considered a necessity by us to meet the complex demands of the application framework. As was already mentioned it was decided to use the Vue.js framework. The decision was made because of Vue’s large library of components by which it can be extended without the need of adapting functionalities to the framework by reprogramming them. New components can also be created and are reusable in every new lesson. Figure 1 demonstrates how the CMS and the Vue.js-based client-side application interact.

Within the vue framework employed by the application the central data object, addressed as “store” by the vue-terminology, represents all information that is loaded from the CMS. Image data that are loaded as GeoTiff is stored in a way that can be referenced by every component within the framework. The parts colored in shades of blue in Fig. 1 represent those components. They can be constantly updated depending on the behavior of the components they are embedded in as well as changes in the store.

All technologies mentioned above resulted in the framework. To assess whether it can technically implement all requirements needed to create applicable lessons, an exemplary lesson, as well as its image processing result, are described and evaluated in Sect. 5.

4.3 Product Comparison

The tools that are implemented in the application framework should resemble their counterparts in professional software but also should be usable without a high degree of prior knowledge so the users can focus on the learning objectives. The cognitive load induced by the operation of the tools in the learning module should be kept as minimal as possible which is why they have to be reduced in functionality. But since the learning modules are meant to offer problem-based learning and the users should use them to answer real-world problems the products derived from the tools have to be comparable to those computed by professional software. To test this, an image classification, as well as a flood hazard map, were produced by different configurations of the application and compared to official data as well as results derived from similar tools using SNAP and QGIS. The SNAP software was chosen for this comparison because it has already been employed in school projects and presents an alternative for educational settings where time constraints are less of an issue and professional tutoring can be provided. The workflows for the assessment of the two datasets are illustrated in Figs. 2 and 3.

The flood hazard map is part of the first prototypical learning unit “Flood Hazard” that is produced with the help of the framework and is also described in more detail in the results section. Users analyze the potential flood risk of an imaginative construction site for a factory in the vicinity of Bonn in Germany. Values below a certain threshold in an 8bit Digital Elevation Model (DEM) of the area can be selected and stored as image layers. The thresholds that are

![Diagram](image.png)
selected resemble the water levels of the Rhine River for three different flood scenarios of this river section. The dataset used in the tool is a resampled subset of LIDAR elevation data provided by the open data portal of the country of North-Rhine Westphalia (IMA GDI.NRW 2022). The official flood scenarios HQ-frequent, HQ-100, and HQ-extreme model the flooded areas in case of flood events that statistically happen once within 10–50 years, once every 100 years, and once in over 100 years and are also represented in official maps for flood hazard assessment (MULNV 2022). The levels at which these scenarios happen were calculated using the online application “FLYS” which is also used by the Federal Institute for Hydrology to model water levels at different discharge rates for every location along major rivers and streams in Germany (Grätz et al. 2015). Aside from a simple visual comparison between officially available flood hazard maps and the ones produced by the tool, difference images were produced based on official flood scenario data. The data was also obtained from the open data portal of the country of North-Rhine Westphalia (IMA GDI.NRW 2022). The three flood scenarios were downloaded from the tool as Portable Networks Graphics (PNG) (Randers-Pehrson 2003) files since it is not able to export GeoTiff Files. Each image represents flooded and non-flooded areas as binary information. They were georeferenced using the extent of the subset of the DEM data. The official data are comprised of elevation data for each scenario ranging from zero to the maximum flood level of the scenario. Those datasets were mosaicked and a subset was created to match the extent of the study area. The official elevation information was flattened to binary flooded and non-flooded information and resampled to the spatial resolution of the base DEM to also match the flood maps derived from the tool. Each tool scenario was then co-registered to the binary map of its corresponding official scenario and compared to highlight where both datasets correlate positive or negative and where the application tool makes a false positive or false negative prediction on flooded areas compared to the official dataset. The band math expression editor in SNAP was employed for this using the pseudocode shown in Fig. 4.

The application framework uses a simple minimum distance algorithm for classification (Albertz 2016). This has the advantage of being less computation-heavy than a more complex method such as maximum likelihood that would not be feasible because of the limited processing power of the browser environment. Of the possible algorithms used for classification, it is also the one that can be most easily explained because of its simple use of the feature space and therefore serves as a good starting point into classification from a didactical perspective. So apart from being too hard to implement because of the computational restraints of the environment used, other classification algorithms such as SVM (Support Vector Machines) or RF (Random Forest) (Albertz 2016) would always act as a “black box” with no way for the pupils or students to easily comprehend what makes a classification possible. The application so far is limited to the use of 8 bit integer values. For ease of use, the training data of the different surfaces are not sampled using pixel data from within polygon boundaries but by placing flags in the areas and thus sampling single-pixel information from each band available in the dataset. Figure 5 illustrates

```plaintext
IF img_tool == neg AND img_official == neg
OUTPUT ("negative")
ELSE IF img_tool == "pos" AND img_official == "neg"
OUTPUT ("false positive")
ELSE IF img_tool == "neg" AND img_official == "pos"
OUTPUT ("false negative")
ELSE IF img_tool == "pos" AND img_official == "pos"
OUTPUT ("positive")
```

Fig. 4 Pseudocode of the expression used in the band-math calculation to determine over- and underprediction of flooded areas.
this simplified process. This way a result can be quickly obtained at the expense of accuracy. To see if the results can still be meaningfully interpreted a subset of a Landsat image of the Aletsch glacier region in Switzerland measuring $1.335 \times 871$ pixels and was reduced to red, green, blue and infrared information was classified into three classes: snow and ice, vegetation, and bare soil. The selection of classes already proved applicable in a school project where the retreat of the Aletsch glacier was monitored using the same image processed in the ESA SNAP software. The coordinates of the training pixels were documented. The image was downloaded from the application in PNG format and georeferenced using the extent of the original Landsat image.

Since the PNG format is only capable of storing color information and transparency, distinctive class information had to be reintroduced using the k-means cluster analysis for an unsupervised reclassification in the SNAP software. The Landsat image was then classified using the documented pixel locations as training data and the minimum distance classifier provided in SNAP. A third classification was produced using the maximum likelihood algorithm in SNAP and training data from polygons and severed as a reference since it should yield a superior classification result. All three products were exported as GeoTIFF for accuracy assessment using the semi-automated classification plug-in in QGIS (Congedo 2021). For the accuracy assessment, 300 points were randomly created and assigned to the classes using the original image. They were as well exported to Google Earth for a more accurate impression of the ground truth where the Landsat image could not be interpreted clearly. The ROIs (Region of Interest) were consistent in each accuracy assessment and were well distributed amongst the classes. The sample size was in line with the literature suggestion of 60 per class (Richards and Jia 1999) with 117 points representing vegetation, 82 bare soil and rock, and 98 for snow and ice.

### 5 Results

This section shows how the described technologies are combined into a working educational application and presents the results of the comparison between data produced with

![Fig.5 Steps of the simplified classification process in the application. Naming a training class, assigning a color, and placing flags in the image to collect spectral information (A); after a minimum of two training classes have been created the classification can be started (B); the classified image is added as a product to the list of available rasters (C)](image)
the help of tools embedded in this application and products derived from software and officially available data products.

5.1 Exemplary Lesson

The prototypical lesson that was created in the system presents a fictional real-world problem. A factory is planned at a location close to the river Rhine because of its advantageous position regarding traffic and transport. The learners are asked to evaluate the flood hazard situation at the location. They learn about flood risk and flood hazards and how earth observation data are used to assess the situation in the eLearning part of the module. The basic structure and idea for this teaching unit was already implemented in the former learning modules and targeted at grade 7–9 of German high schools. As a prototype lesson, it meets all the demands that are formulated in Sect. 2.

In the content management system, all the chapters and the respective contents are defined. So it will serve as a starting point to demonstrate how lessons are constructed. The basic entity to store information in WordPress is a post and every lesson is represented as a post. Within the posts, a system called “Gutenberg Blocks” (WordPress Foundation 2022a) can be used to freely combine different content types into a concise information page. These blocks cover the basic types like text, headlines or images. Custom blocks were created to accommodate more advanced types, i.e. the upload and definition of a dataset that is to be used in the lesson. Those custom blocks contain the configuration information needed by the application framework and can be created using JavaScript. The blocks are ordered in a chapter structure, which also represents the structure of the learning module. The lesson “Floods” is divided into four chapters. The eLearning part of each chapter consists of different types of media for conveying background information on the topic. The image processing part of a chapter can contain different datasets that are to be used for image processing. Both parts can be accessed and switched between at any time (see Fig. 6).

What kind of processing can be done is also defined in the chapter. Specific tools can be associated with the data, but there is also a set of primary tools like adding markers or reading pixel information, that is available to all the datasets in the chapter. Both tool categories are thereby specific to either a dataset or a chapter and are implemented in the CMS using custom blocks. This way the available toolset is always tailored to the needs of the lesson and thus lowering the extraneous cognitive load by not exposing the learner to an excessive amount of functionalities but rather increasing the complexity step by step. Chapter one requires no work with the image processing or the map component so no data or tools are defined on the CMS. Here only the task, information material, and a quiz are defined. The Quiz is used to separate the first chapter from the second since every chapter should not be available from the start. The image processing in this exemplary lesson is based on a DEM that is implemented in chapters two and three. In chapter two, no tools are associated with the dataset but students are asked to mark certain points on the image and measure their elevation. Thus only the pipette and the point marker are available as primary tools. Chapter two shows how background information can be directly interwoven with the application of the information (see Fig. 7).

Here an animation of a plane scanning a strip of land and acquiring height information is shown alongside the resulting DEM illustrating, how the dataset that is used was produced and how it can be interpreted. In the application part, this knowledge can be directly put to use by working with DEM data. In the third chapter, the DEM can be employed to answer the initial question of what level of flood hazard the planned factory is exposed to. A tool was created to mark all values below a certain threshold in the image data. Figure 8 shows how the data are implemented in the CMS and presented in the application.
The students start at a level of 46 m above sea level and are asked to create three flood scenarios using a slider and thereby raising the level by 6 m, 8 m and 10 m which roughly corresponds to the official scenarios for flood hazard assessment. Each result of the threshold analysis can be saved to the dataset and different colors can be assigned. Changing the opacity or the layers or their display order then allows users to analyze which flood scenario could potentially put the factory at risk. How well this quickly produced flood hazard map corresponds to official scenarios is shown in the second part of this chapter. Quizzes, that can only be solved by producing and interpreting the flood hazard map help to separate the chapter from the final chapter where the topic of flood hazard and flood risk to the human sphere is wrapped up. Here another important content type is implemented. An interactive animation produced in HTML5 and JavaScript shows how different anthropogenic and natural factors influence the flood hazard. This also uses a custom block in the CMS. Figure 9 shows how this chapter is represented in both CMS and application.

5.2 Result Comparison

Given the correct use of the tool and the numbers given in the task, students can produce a flood hazard map and base
their conclusions on it. This product is used to answer the first research question. A comparison between the results achieved with the tool and the official map provided by the ministry of environment, agriculture, conservation, and consumer protection can be seen in Fig. 10.

Apart from this visual comparison, the dataset from which the official map was derived was used for a more in-depth comparison. The images in Fig. 11 shows where differences occur. Green colors indicate true positive pixels that are estimated as flooded areas by the tool as well as the official data. Red pixels show areas that the official data labeled as “flooded” but the tool failed to predict. Yellow areas show the opposite where the tool falsely predicted flooded areas, black pixels are areas where neither the tool nor the data mapped potentially flooded areas. The information in both datasets for those areas was “noData” represented as zero in the image.

Table 1 quantifies the visible differences in the images of Fig. 11 as confusion matrices for each scenario, including the sensitivity as the probability of predicting a positive match, as well as the specificity as the probability to predict a negative match.

Even though it is not yet integrated into an applicable lesson the outcome of the classification tool was also compared in terms of accuracy to products produced by software to answer the second research question. Figure 12 is showing the results of the application and classifications produced by the ESA SNAP software. The accuracies for each classification can be seen in Table 2. Here the accuracies for all classes of each classification are given alongside the overall accuracy of the respective classification as well as the kappa values indicating in how far the classification results differ from random assignment of classes (Hudson and Ramm 1987).

6 Discussion

The exemplary lesson has shown, that the technology stack described in Sect. 2 can be meaningfully combined into a framework containing content creation and application that meets the requirements needed to implement problem-based learning in a school environment on a technical level. A CMS like WordPress can be employed to host the information on most types of content that are needed in an eLearning...
environment for remote sensing education. The fact, that WordPress is highly modifiable also allowed for the creation of content types specific to our application scenarios like the upload of GeoTiff data and tool configuration. In the Front End Application, the use of a modern development framework like “Vue.js” allows the use of an infrastructure that makes constantly updated information accessible to multiple components of the application which results in a dynamic behavior and also synchronizes the learning content of each chapter with the availability of data and tools. Also, the component-based environment includes the possibility of creating “single file components”. This makes the system easily expandable since the tools can be developed and modified in a way not unlike an object-based style of programming. The additional libraries “GeoTiff.js” as well as “Geoblaze.js” needed for image display and processing could be integrated into the “Vue.js” and Open Layers could also be integrated into a vue.js single file component.

The processing results achieved by the application’s tools show, that the simplification of data and tools for image analysis can yield reasonably good results. In the flood hazard map that can be produced with the tool in the exemplary lesson the location for which the flood risk is to be assessed lies within the HQfrequent flood scenario. So both the official data as well as the map, produced with very little effort, are pointing to the same conclusions. So the first research question:

“Does a flood hazard map produced with tools reduced in complexity generate a result comparable to an official one?”

![Fig. 10](https://www.uvo.nrw.de/)

![Fig. 11](https://www.uvo.nrw.de/)

Fig. 10  Visual comparison of the flood hazard map created by the tool to the official flood hazard maps (https://www.uvo.nrw.de/)

Fig. 11  Difference images showing over- and underpredicted areas. A and B mark areas of strong false positive overprediction.
cial flood hazard map?” was positively confirmed for the condition that: “The visual conclusion of a site analysis conducted in a problem-based learning scenario in the environment leads to the same conclusion as an openly accessible flood hazard map”.

The second condition: “In a statistical comparison of both flood hazard maps, the flooded and non-flooded areas in the hazard map derived from the tool predict the flooded and non-flooded areas of official data to a percentage of 80% and higher.” could also be confirmed.

The results in Fig. 11 show that the scenarios predicted by the application tool tend to yield falsely positive results and thereby over-predict the flood scenario. This is apparent in two major cases in the areas marked “A” in the “Frequent” scenario and “B” in the “Extreme” scenario. The latter can be explained by a stream that is not included in the official comparison data used for the accuracy analysis since it was not available at the time. Area “A” is most likely the result of the change in datatype from float32 to unit8 and the resulting reduction in bit depth for the DEM used by the tool. In the tool students can only separate threshold values in meter but not in sub-meter dimensions. This is of course bound to produce a certain inaccuracy in the resulting flood hazard map since differences in centimeters of the water level can have a severe impact on the flood situation. Still, the high sensitivity values of the confusion matrices shown in Table 1 support the visual impression of a high degree of similarity between the compared datasets. In both the “HQfrequent” and “HQ100” scenarios the probability of the simple threshold classification tool of the application predicting a positive match with the official dataset is well above 85%. The lower value of 80% for the “HQextreme” scenario can also be explained by the mismatch of the datasets that led to the false-positive predictions of area B in Fig. 11. Given the intended application of this tool for educational purposes, a sensitivity between 80 and 89% seems sufficient to create a product that closely resembles the official analysis of flood hazards in the area. Since the vast majority of the pixels in both datasets are “noData” values resulting in negative

### Table 1 Binary confusion matrices showing the prediction accuracy of the tool in comparison with official data as reference

|                | Tool positive (PP) | Tool negative (PN) |
|----------------|-------------------|-------------------|
| HQfrequent     |                   |                   |
| Data positive P | 100324            | 6289              |
| Data negative N| 15140             | 1406511           |
| Sensitivity    | 0.86              |                   |
| Specificity    | 0.99              |                   |
| HQ100          |                   |                   |
| Data positive P | 131999            | 4662              |
| Data negative N| 15782             | 1297735           |
| Sensitivity    | 0.89              |                   |
| Specificity    | 0.99              |                   |
| HQextreme      |                   |                   |
| Data positive P | 162215            | 2543              |
| Data negative N| 39270             | 1324236           |
| Sensitivity    | 0.80              |                   |
| Specificity    | 0.99              |                   |

Crucial accuracy parameters are marked bold

![Classification results. A Original Image as RGB composite with locations of ROI data for accuracy assessment visible in red. B Minimum distance classification results of the tool embedded in the application; C minimum distance classification result achieved with the same training data in SNAP; D maximum Likelihood classification result in SNAP](image)
prediction the specificity for all scenarios is very high at 99%.

A certain inaccuracy in the tested classification tool was expected and did not happen at a level that would prevent students from drawing conclusions based on the products they created. The main source for the inaccuracies in the classification is also most likely the conversion of the original bit-depth of 16bit depth of the Landsat data to 8bit for the image used in the tool. In the classification, this reduced radiometric resolution can have an impact on accuracy when separating classes of a certain spectral similarity like snow and ice and rock that both have high reflectance characteristics across the RGBIR range, especially in the fringe areas of snow cover where mixed pixels occur. But even given those drawbacks in radiometric resolution the overall accuracy of 78.66 for the map produced by the tool is comparable and statistically even slightly superior to the minimum distance classification conducted in the SNAP software (see Table 2) with an accuracy of 73.33. So the second research question: “Does a classification produced with tools reduced in complexity generate comparable results to professional software?” under the condition: “The overall accuracy of the classification tool matches the accuracy achieved with the same algorithm in a professional software.” was also answered positively. The overall accuracy of 89.66, as well as the high Kappa of 0.84 of the maximum likelihood classifier, reinforce the assumption of it being the most accurate classification of the scene and it therefore can serve as a reference. While statically inferior to the classification performed in the tool, the visual impression puts the results of the SNAP minimum distance classifier closer to the output of the maximum likelihood classifier. This contradiction can be attributed to the randomly created ground control points for accuracy assessment that, even being numerous and well distributed amongst the classes, leave room for errors.

The design of the tools that prioritize ease of use over accuracy can also be seen as a reason for the difference in the performance of the application tools in comparison to the SNAP software. This is apparent in the classification example that uses a simple minimum distance classifier which may be seldom used in real-world scenarios but, because of its simplicity, is a perfect starting point for explaining the concept of classification in general. The sampling of the spectral characteristic for the training data through point data would also not be considered sufficient in normal scientific or applied use cases, but since achieving visible results in a considerable amount of time and avoiding frustration is considered a priority, the manageable trade-off in accuracy seems an acceptable price to pay.

The hypothesis that this work was based on holds true. **Browser-based tools for educational remote sensing image analysis with a reduced complexity in terms of tools and data can provide results that are sufficiently accurate to represent real-world application scenarios.**

The developed learning environment is capable of providing simplified methods for image analysis that can potentially be employed in problem-based scenarios using real-world examples.

### 7 Conclusion and Outlook

The technical evaluation of the developed framework shows how image processing can be made more easily accessible. It enables the content creator to deliver a concise package including teaching materials and background information in different multimedia formats. The utilization of a web development framework in combination with a CMS enabled allowed to create a framework that offers simplified handling of tools for image analysis while meaningfully combining this hands-on experience with a scaffold of chapters, tasks, and background information. While the framework is not yet actively used in schools and other educational environments it has been shown that it is capable of delivering results that would allow pupils to draw conclusions about the state and processes of the earth’s surface from self-produced information on a technical level. By relying on client-side image computation, it has also demonstrated, that easily accessible image processing is not restrained to a pure web environment. While in its current state it still depends on a server-side delivered object containing its structure and datasets, it
leaves the opportunity to combine all assets into a desktop application and compile it for any operating system using, i.e. the electron (GitHub Inc. 2013) technology which has already been tested in prototypes of the framework. The downside of this is the dependency on JavaScript as a programming language that was not intended for this kind of use. Larger image data or more complex image processing is still only possible in server-side systems or professional desktop software. But in the light of the system’s focus on educational use and the possibilities for offline usage, this seems a fair compromise. One has to consider, that education on earth observation not only happens in areas that are well connected to the internet. Also, technological advances in web development are already foreshadowing developments that will equip the client-side with considerably more processing power by making virtually every programming language available in the browser through the Web Assembly technology making it for example possible to access GDAL functionalities in the Browser (Dohler 2021). After the technical evaluation of the framework, the next step will be testing the setup in actual educational scenarios. As mentioned in Sect. 1, the overarching concept of the framework is based on proven findings gathered from past learning modules and did not change but was rather improved by a more elaborated user interface of the applications. So the applicability itself is mainly a question of whether the interaction of the user with the application is satisfying or needs further refinement. Since the GUI (Graphical User Interface) is designed using the Vue.js framework, modifying it by making changes to the GUI is achievable with considerable low effort. A larger concern for the applicability in schools is the content and how the materials can be adapted to the curricula. Experiences drawn from similar applications created in the past have already shown the advantages of this approach (Goetzke 2014; Hodam et al. 2020). But this has ultimately to be proven in an empirical study in which groups are exposed to the teaching material and their learning progress, as well as aspects of user experience, can be tracked and analyzed. This work laid a sound foundation for this future investigations, by ensuring the technical capabilities of the system. Since the effort of creating and modifying content has been greatly reduced by the newly developed framework there is now even more opportunity to realize efficient learning and teaching materials since the workload can be shifted from technical development to content creation.

Acknowledgements The learning environments described in this article are part of the Remote Sensing in Schools Project. The project is funded by the German Aerospace Center (DLR) with interruption since 2006. Its primary goal is to find ways to integrate earth observation topics in school classrooms for all STEM subjects. It also represents a way of distributing basic knowledge of earth observation to the interested public as well as university students. While the integrated learning environments described in this article play a crucial part in this effort, they are embedded in an online environment that also consists of background information and additional classroom resources. The projects also carries out teacher trainings as well as hands on projects with pupils. The development is also part of the efforts of the EO College Project to establish an infrastructure for education in remote sensing.

Author Contributions Conceptualization: Henryk Hodam; methodology: Henryk Hodam; software: Henryk Hodam; validation: Henryk Hodam; formal analysis: Henryk Hodam; investigation: Henryk Hodam; resources: Henryk Hodam; data curation: Henryk Hodam; writing—original draft preparation: Henryk Hodam; writing—review and editing: Henryk Hodam, Andreas Rienow and Carsten Juergens; visualization: Henryk Hodam; supervision: Andreas Rienow and Carsten Juergens; project administration: Andreas Rienow; funding acquisition: Andreas Rienow and Carsten Juergens. All authors have read and agreed to the published version of the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. This research was funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) through the German Aerospace Center (DLR). (DLR) Grant numbers: 50 EE 1703 and 50EE1924B. We acknowledge support by the DFG Open Access Publication Funds of the Ruhr-Universität Bochum.

Declarations

Conflict of interest The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Adobe Flash. Adobe Inc. (2021) https://www.adobe.com/de/products/flashplayer/end-of-life-alternative.html#test. Accessed 1 Feb 2022
Albertz J (2016) Einführung in die Fernerkundung.: Grundlagen der Interpretation von Luft- und Satellitenbildern. Wiss. Buchges, Darmstadt
Amici S, Tesar M (2020) Building skills for the future: teaching high school students to utilize remote sensing of wildfires. Remote Sens 12:3635. https://doi.org/10.3390/rs12213635
Asimakopoulou P, Nastos P, Vassilakis E, Hatzaki M, Antonarakou A (2021) Earth observation as a facilitator of climate change education in schools: the teachers’ perspectives. Remote Sens 13:1587. https://doi.org/10.3390/rs13081587
Buyaert D (2022) Drupal. PHP. Buyaert, Dries
Congedo L (2021) Semi-automatic classification plugin: a Python tool for the download and processing of remote sensing images in QGIS. JOSS 6:3172. https://doi.org/10.21105/joss.03172
Rienow A, Goetzke R, Hodam H, Voß K (2013) Das FIS-Lernportal - Beiträge zur kompetenzorientierten Bildung. Computer + Unterricht
Rienow A, Lindner C, Dedring T, Hodam H, Ortwein A, Schultz J, Selg F, Staar K, Jürgens C (2020) Augmented reality and virtual reality applications based on satellite-borne and ISS-borne remote sensing data for school lessons. PFG 88:187–198. https://doi.org/10.1007/s41064-020-00113-0
Santillan D (2022) plotly.js. EOX IT Services GmbH. https://github.com/santillan/plotly. Accessed 22 Jan 2022
Savery JR, Duffy TM (1995) Problem based learning: an instructional model and its constructivist framework. Educ Technol 35:31–38
Schindler F (2022) geotiff.js. EOX IT Services GmbH. https://github.com/geotiffjs/geotiff.js/. Accessed 22 Jan 2022
Schunk DH (2019) Learning theories: an educational perspective, 6th edn. Pearson, Boston
Siegmund A (2011) Satellitenbilder im Unterricht. Heidelberg, Pädag. Hochsch., Diss., Pädagogische Hochschule Heidelberg
Sinergise Laboratory for Geographical Information Systems, Ltd. (2017) EO Browser. https://apps.sentinel-hub.com eo-browser. Accessed 20 Dec 2021
SNAP. ESA. (2022) https://step.esa.int/main/download/snap-download/. Accessed 4 Dec 2021
Song J, Zhang M, Xie H (2019) Design and Implementation of a Vue.js-based college teaching system. Int J Emerg Technol Learn 14:59. https://doi.org/10.3991/ijet.v14i13.10709
Sweller J (1988) Cognitive load during problem solving: effects on learning. Cogn Sci 12:257–285. https://doi.org/10.1207/s15516709cog1202_4
Syafiq S, Daud M, Hafizah H, Ahmad Z, Imri S, Akmar E, Rahim N (2018) Comparison of web development using framework over library. Int J Inf Control Comput Sci. https://doi.org/10.5281/ZENODO.1316514
Terhart E (2003) Constructivism and teaching: a new paradigm in general didactics? J Curric Stud 35:25–44. https://doi.org/10.1080/00220270210163653
Torchbox (2022) Wagtail CMS. Python. Torchbox
Tsou M-H, Guo L, Howser A (2005) A web-based java framework for cross-platform mobile GIS and remote sensing applications. Gisci Remote Sens 42:333–357. https://doi.org/10.2747/1548-1603.42.4.333
Voß K, Hodam H, Goetzke R, Rienow A, Gutenberg U (2013) Praxisbeispiel: Kurzvorstellung: Vom Satellitenbild zur Karte: Interaktiv Geomethoden lernen. Computer + Unterricht
WordPress Foundation (2022a) Block editor handbook. https://developer.wordpress.org/block-editor/. Accessed 17 Feb 2022
WordPress Foundation (2022b) Wordpress. WordPress Foundation. https://wordpress.com/. Accessed 17 Feb 2022
Yao Z, Ma Z, Liu Y, Amiri Sani A, Chandramowlishwaran A (2018) Sugar: secure GPU acceleration in web browsers. In: Shen X, Tuck J, Bianchini R, Sarkar V (eds) Proceedings of the twenty-third international conference on architectural support for programming languages and operating systems. ACM, New York, pp 519–534
You E (2019) vue.js. https://vuejs.org/. Accessed 22 Jan 2022