

**Patricio Becerra**

Grüneckweg 10B, CH-3006, Bern  
patriciobecerrav@gmail.com

## **Data Science Project**

# **Image-based Classification of Sediments in Drill Cores**

## **Conceptual Design Report**

**15 December 2023**

### **Abstract**

Subsurface sediment cores contain a great deal of information about the geologic and climatic history of a region. The stratigraphic models derived from these cores are invaluable tools in landscape assessment within a variety of scientific and applied fields, from geophysics to environmental modeling, to land and resource management. These stratigraphic models are constructed through visual analysis of the core with the goal of identifying and classifying the various lithologies present in the subsurface, a process that can take several months and requires sedimentological expertise. This project aims to build a Deep Learning model that can automatically classify the sediment classes using scanned images of the core, reducing the analysis time to a few hours, and setting the foundation for generalizable classification models that can classify the subsurface layers of any type of subsurface core.

# Content

Abstract	0
Content	1
1 Project Objectives	2
2 Methods	3
3 Data	4
4 Metadata	6
5 Data Quality	6
6 Data Flow	7
7 Data Model	8
8 Documentation	9
9 Risks	9
10 Preliminary Studies	11
11 Conclusions	12
Acknowledgements	12
Statement	12
References	13

# 1 Project Objectives

The analysis of subsurface cores is a fundamental process in the geological sciences. In a general sense, core-drilling entails the extraction of an intact sample of the subsurface (known as a “drill core”) from a specific area of geologic interest. In this core, samples of past surface layers are preserved, from which the geologic and climatologic history of a region, continent and even the whole planet can be retrieved and utilized to develop models that address a variety of environmental and societal issues. Depending on the scientific objective and geographic location, cores can be anywhere from tens to thousands of meters in depth, with the deeper layers correspond to older geologic periods.

The subsurface layers seen in a drill core can be classified by shared physical or chemical properties, which often indicate a common formation environment. Sedimentary layers – or “facies” – can be classified by their lithoclastic type, which is related to the average grain size of the sediment (e.g. sand vs. gravel in the case of unconsolidated sediments). This classification is done through visual analysis, requires sedimentological expertise, and often takes several months per core to complete. Part of the core curation and preparation for analysis requires the scanning and data-logging of the core, in which high-quality images, as well as important geophysical data such as density or magnetic susceptibility, are captured.

In the context of his doctoral studies, ADS CAS co-student Sebastian Schaller has participated in a core-extraction and analysis campaign in the Northern Alpine Foreland [1–3], in which 11 cores were extracted to study the effects of the many cycles of glacier advance and retreat during the Middle-to-Late Pleistocene<sup>a</sup> epochs. Though the analysis and litho-classification of these cores has taken Sebastian and his colleagues ~6 months, the scanning itself takes only a few hours. The goal of this project is to **build an algorithm that can be trained on the scanned core images to perform the visual classification<sup>b</sup> automatically**. Ideally, the algorithm will also produce a stratigraphic model or column (i.e., plots of depth vs. lithoclastic type) that can be compared with the visual classification performed previously, allowing for a proper quality assessment. If successful, this algorithm would reduce the analytical workload by orders of magnitude and optimize the development of stratigraphic models, from which the major scientific value of drill-cores is obtained. In addition, this approach could be combined with geophysical data-based clustering (S. Schaller’s CAS project [4]) for further optimization and advantages.

Similar algorithms have been developed in the last few years with good results within the local regions of study. Some have focused on the geophysical data [5,6], and some have used images directly [7-10]. One recent study used semantic segmentation to put forth a more generic, less geographically constrained approach for the first time [11]. I plan to heavily reference these studies as needed during this project.

The full power of an algorithm capable of automatically classifying the Alpine foreland sediment cores lies in its potential to be generalized to perform the same task on any type of drill-core, from deep-sea sediment cores [12] to ice sheet cores [13] to, eventually, cores from the Moon [14] or Mars [15,16].

---

<sup>a</sup> The middle-to-late Pleistocene epochs comprise the period between ~800'000 years ago (ka) and ~12'000 ka.

<sup>b</sup> It is important to distinguish sedimentary classification from *classification* in the ML sense. The latter usually refers to a specific class of supervised ML algorithms. However, for this project, I plan to use unsupervised clustering ML methods to “discover” classes of sediments in the core, and only compare them with the visual interpretation after the automated clustering to evaluate the success of the algorithm.

## 2 Methods

The infrastructure I will use for the project is an Anaconda distribution of Python installed locally under a project-specific environment, and interactively used on Visual Studio Code (VSC). I will use Jupyter notebooks in VSC for data preparation, preliminary analyses, and for prototyping code. I expect to migrate the code for the final algorithm to standard Python (.py) files for easier application.

As for the specific libraries and tools that I will use, it is not possible to know everything at this stage, but I expect to use at least the following Python libraries:

### Data handling

- OS [17] – Miscellaneous operating system interface and commands
- Pandas [18] – Dataset manipulation and import/export of .csv files and/or spreadsheets
- Numpy [19] – Numerical computing and multidimensional array manipulation
- Scipy (primarily scipy.stats) [20] – Scientific computing, statistical methods and analyses
- Statsmodels [21] – Statistical models and tests

### Image manipulation and analysis

- Scikit-image [22] – Image processing
- Pillow [23] – Image manipulation and processing
- OpenCV [24] – Image segmentation and object detection

### Machine Learning

- Scikit-learn [25] – Machine Learning (ML) and predictive data analysis
- TensorFlow [26] – Deep Learning (DL) and Convolutional Neural Network (CNN) functionalities

### Data visualisation

- Matplotlib.pyplot [27] – Display and plotting.
- Mpltern [28] – Plotting ternary plots (if necessary).
- Seaborn [29] – Statistical data visualisation
- Plotly [30] – Data analytics plotting

The sediment cores have already been scanned and the images preliminarily processed. Scans of one of the cores (site 5068\_2A) were used for the visual litho-classification published in [1]. A substantial portion of this data is publicly available from the ICDP-DOVE (International Continental Scientific Drilling Program – Drilling Overdeepened Alpine Valleys) project [2] website [3]. In addition, non-public data is available through collaboration with S. Schaller, who is a direct participant in the DOVE project.

Although most images are of similar high quality, they have slightly different dimensions, and may include more than one type of sediment type in each file. Therefore, the images will have to be split, and more importantly, labeled according to the lithotype. According to [1], up to 14 lithotypes could be identified, but the range of lithotypes is relatively continuous, such that some types are quite similar to

each other. Therefore, for a first approach, the four types to be considered will be: Diamict, Gravel, Sand, and Fines (see next section).

The image data can be combined with the Multi Sensor Core Logger (MSCL) data (density, magnetic susceptibility, natural gamma radiation [1,4]) prior to the ML model training, or it can be treated independently, such that the model would be trained on image data only. The former approach would most likely produce more complete results for the local region of interest (ROI), while the latter sets the foundation for a generic image-based sediment recognition model. Both approaches will be attempted.

After the data is prepared, standard statistical analyses can be performed to learn about the image data, and further clean it. E.g. there are certain sections of the core where the quality of the core material was not high enough, or where the core itself was destroyed. In these cases it is not possible to keep a high quality image. The MSCL log data can be used to assess which sections can be ruled out. The data will also have to be normalized (i.e., the pixel values require normalization) so that the ML techniques applied can function properly.

With the cleaned and labeled image dataset, the application of Machine Learning models can begin, with the goal of finding the model that is best able to reproduce the visual classification. An additional goal, related to the goals of [4], is to find patterns in the data and see how well these patterns correlate to the visually classified lithotypes. Both supervised and unsupervised methods could be attempted.

### 3 Data

The data consists of images of at least 2, and at most 11 sediment cores scanned during the DOVE project [29]. The study area is in the southwestern part of the former Rhine Glacier, close to the city of Schaffhausen in northern Switzerland [1]. Each core is ~150 – 200 m in depth, and the 11 drill sites amount to a total of 1800 m of core material. There are two drill sites for which the data has been made fully public [3]; these two sites (site 1C and site 2A) have 199 and 278 images respectively, for a minimum total of 477 images. The data for the other 9 sites is not public but can be accessed through collaboration with S. Schaller. Each scanned image covers roughly 0.5 – 1 m of the core, so that a conservative estimate for the maximum number of images available is ~ 1800.

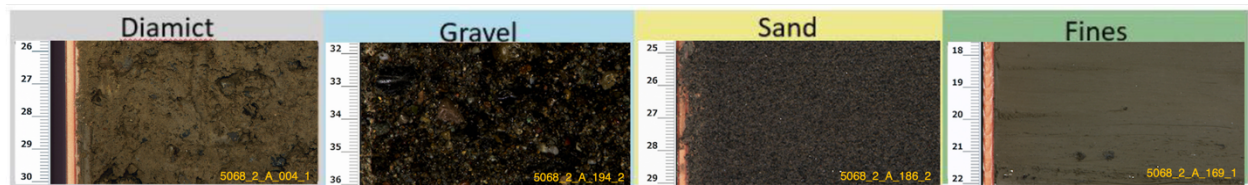


**Figure 1.** Image CS\_5068\_2\_A\_169\_1\_A.jpg of a 0.51 m section of the core taken from site 2A. Top depth: 177.5 m.

Figure 1 shows an example of a typical image file, taken from a 0.51 m section of core 2A (for which the stratigraphic model was published by [1]), at a depth of 177.5 m below the surface. There are mostly

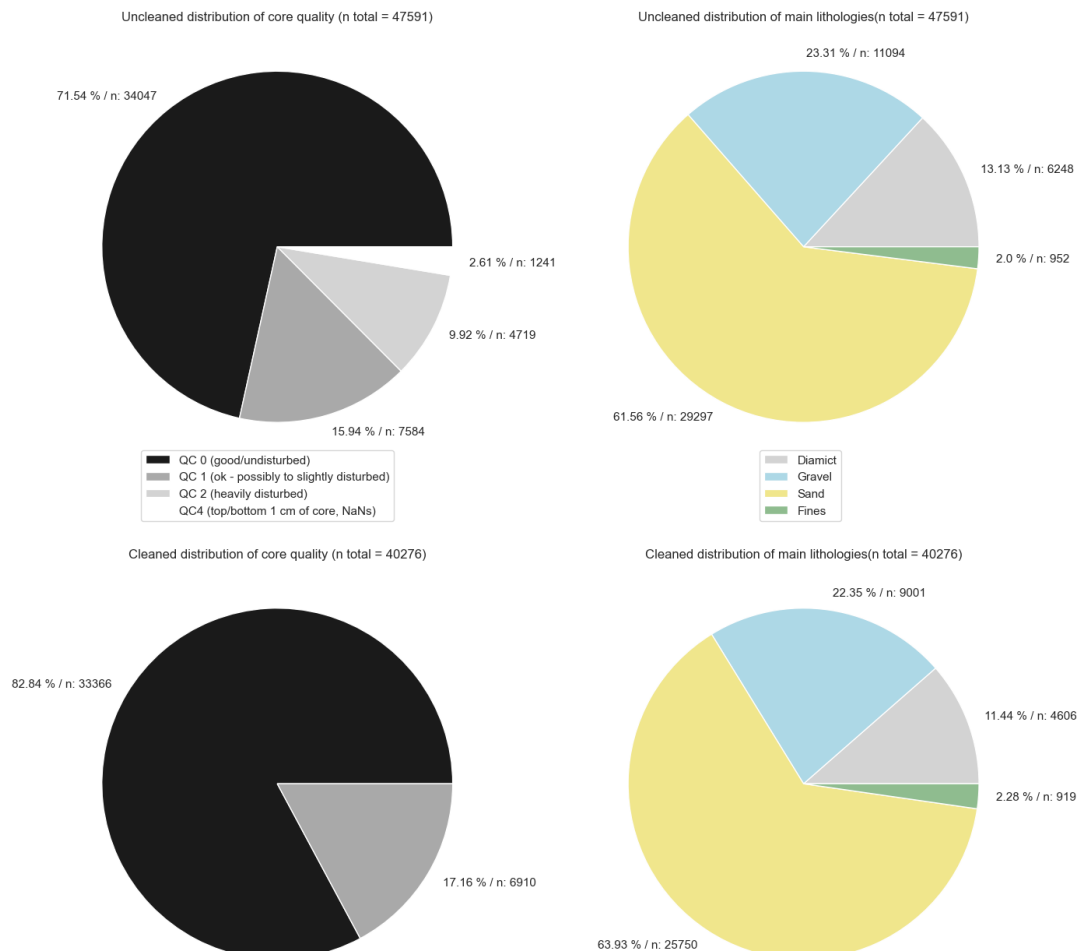
two lithotypes visible in this section: Sand at the top 15 cm, and mostly Fines with some interspaced sand layers in the rest of the section.

Figure 2 shows the four major lithological types identified by [1], and which will be used to label the images accordingly. These types are present in all 11 sites and are therefore representative of the major lithotypes encountered in the study area. Note that, at least for the public data, labeling is straightforward since the visual classification has already been performed and published in [1].



**Figure 2.** Examples of the four main lithologies observed in the drill core of site 2A, but applicable to all 11 cores extracted from the ROI [1,4].

During Modules 2 and 3 of the CAS, statistical summaries of the MSCL log data from site 2A were made. Figure 3 shows the distribution of the visually classified lithological types in that core, as well as a measure of the amount of the core that can be considered “undisturbed”, or of high quality.



**Figure 3.** Quality (left) and lithology (right) distributions of the MSCL log data for core 2A. The “cleaned” data excludes all NaNs and includes only high-quality (undisturbed to slightly disturbed) core sections.

The statistical summary shown in figure 3 displays the approximate distribution expected to be found by a classification model. About 60% of the core material was classified as Sand, and almost a quarter as Gravel. At this stage of the project, the data in Figure 3 can be considered approximately representative of all cores and thus, of the ROI.

## 4 Metadata

The main piece of metadata that will accompany the images is the image file name (e.g. CS\_5068\_2\_A\_169\_1\_A.jpg), which contains information about the expedition (5068), site (2), hole (A), and core section (169\_1\_A). This file name is directly tied to the corresponding logged section, as explained in section 4 of S. Schaller's CDR [4]. CSV files with all available data on the extracted cores is available from the DOVE project. Aside from the image filename, the metadata for the images is thus essentially the same as that for the MSCL log data being used by [4].

Each core section is assigned a unique combined ID, and the top and bottom depths of each section are assigned after drilling. These depths, along with the date of logging, are also hand-written in each image next to the colour references at the bottom of each image frame (e.g. the notes in Figure 1 record a top depth of 177.5 m, and a bottom depth of 178 m for this section). The filename, section ID, and depths link all data available on that section, i.e. image, visual classification, core samples, and MSCL logs.

All the data (except the raw MSCL data and some metadata used in processing) is being made available in the DOVE project website [3]. At the moment of writing, only the data for the 2 core sites mentioned in section 3 are accessible. All other data will be used through internal collaboration with S. Schaller and the DOVE project and will be made available at the prerogative and schedule of the project.

## 5 Data Quality

In terms of the images themselves, all images are of high and equal quality, as they were all captured via a standardized methodology by the same MSCL scanner. Each image is a JPEG file of roughly 1000×2000 – 9000 pixels and ~ 2–10 MB in file size.

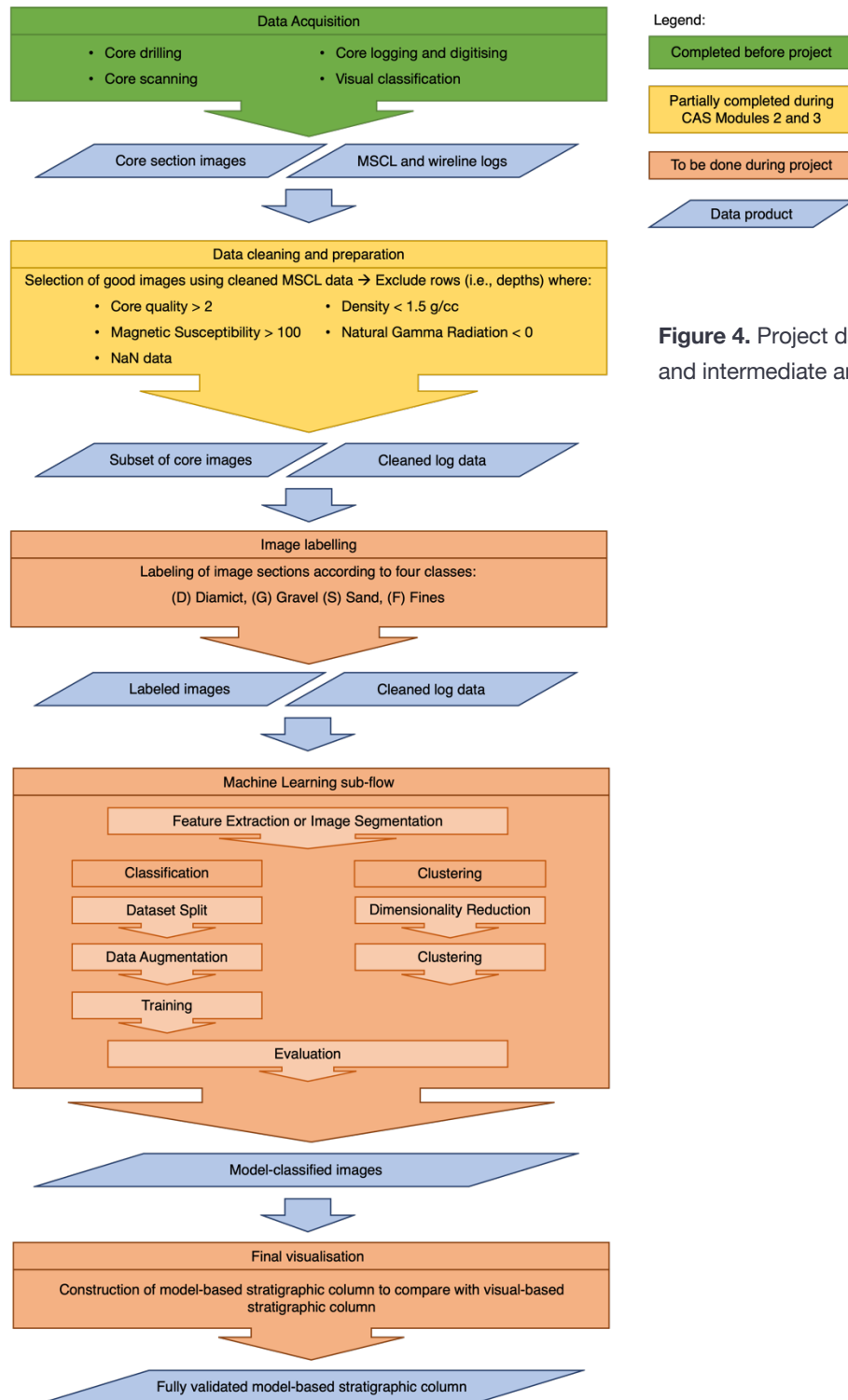
Of the 477 – 1800 images available, not all will be usable, as their quality depends on the quality of the core. This is related to the degree to which a section is disturbed or has lost its structural and physical information during extraction (see [1] and [4] for further details). Figure 3 shows that 84% of MSCL data points are of high enough core quality to be used. So, of the total number of images, 400 will be usable if only the public cores are taken into account, while ~1500 will be usable if all 11 cores are considered.

Independently of quality, the total number of images available could potentially be an issue for a CNN model, as they will perform better with more images in the training set. However, the dataset used by [11] contained only 82 images (covering 31 sites, but with core depths of <50 m), which they divided into 63 for training, 9 for validation and 10 for testing. Given that the minimum number images available for this project is 400, I expect to be able to construct a model that will meet the objectives even in the worst-case scenario of using only 2 cores.



## 6 Data Flow

The project data flow has the following stages or modules: (I) Data Acquisition, (II) Data cleaning and preparation, (III) Image labeling, (IV) Machine Learning, and (V) Final visualisation. Details on each stage are shown on figure 4, and further details on stage (IV) are described below.



**Figure 4.** Project data flow with process stages and intermediate and final data products shown.



In general, the Machine Learning sub-flow (stage (IV)), has the following approach:

- Image Classification (Supervised Learning):
  - (a) Feature extraction/Segmentation model: A pre-trained CNN architecture will be used, fine-tuned to be applicable to the sediment core data. This approach is similar to that of [9], so similar architectures will be attempted, e.g. U-Net [31], EfficientNet [32], or ResNet [33], all available for implementation through TensorFlow [26].
  - (b) Dataset Split: The image dataset will be divided into training and test sets, ensuring a balanced representation of each sediment type in both sets.
  - (c) Data Augmentation: Transformations like rotation, flipping, and zooming will be applied to the training set, to increase model robustness.
  - (d) Training: The model is then trained on the training set and validated on the test set. Metrics such as accuracy, precision, recall, and F1-score can be evaluated here.
  - (e) Evaluation: Finally, the model is evaluated on the test set, using a confusion matrix and visual comparison of the model classification vs. the visual stratigraphic classification. A model-based stratigraphic column can be produced for comparison.
- Image Clustering (Unsupervised Learning):
  - (a) Feature extraction/Segmentation model: As above, a pre-trained CNN will be used to segment and extract features from the core images.
  - (b) Dimensionality Reduction: Dimensionality reduction techniques (e.g., PCA) can be used to reduce feature space.
  - (c) Clustering: Algorithms such as K-Means and HDBSCAN would then be used to find groups of similar images.
  - (d) Evaluation: The quality of the clusters will be assessed by visual inspection, i.e., by comparing a clustering-based stratigraphic column to the column created during the visual classification of the sediments.

A hybrid method, wherein the MSCL log data features [1,4] are combined with the image features, can be embedded in either of the two algorithms above. Whether or not this methodology is implemented will be decided once the project has matured.

## 7 Data Model

The data model can be defined within three levels:

At the conceptual level, the model is a machine-learning tool that can perform a reliable classification of sediments within a drill-core. At minimum, the tool should perform well for test data taken from the ROI. At most, it should be the baseline for a model that could be generically applied to any drill core.

At the logical level, the model will be based on pre-trained CNN's to segment and extract features from the images, after which two parallel approaches, supervised classification, and unsupervised clustering, will be used to automatically determine the lithological type of an image or a section of an image and build

a stratigraphic model. The lower-level-features are the pixel brightnesses in the R, G and B channels of each image, used in order to find higher-level features, such as grain size, colour, presence of clasts, etc.

On a physical level, I expect to be able to complete stages (II) and (III) of the Data Flow using a conventional laptop or desktop computer. The infrastructure necessary to run the actual training during the DL stage (IV) depends on the specific CNN architecture used, but it will probably require access to a high-performance GPU. For this purpose, cloud services like Google Colab – which provide access to GPU-enabled virtual machines for training DL models – are likely to be necessary. In addition, pre-trained models (e.g. ImageNet [34]) will be used to optimise performance with limited resources.

## 8 Documentation

I plan to perform most of the initial tests and code prototyping in well documented and commented Jupyter notebooks. The code in the notebooks can then be migrated to .py for faster application and for more efficient function definition and use. Furthermore, the markup in the notebooks will be the basis for readme files that will assist the user in the application of the model. Finally, all version control and final version storage will be done using Git and stored in an online Git repository through GitHub (<https://github.com/pbecerrav>).

## 9 Risks

At the stage of this CDR, I have identified four distinct risks:

- (R1) *Difficulty in image segmentation and clustering due to the similarity of subclasses.*** As can be seen in figure 1 and is explained in [1], the spectrum of sedimentary lithological types is not entirely discrete, but rather continuous. Classes such as “sandy clay” (e.g. lithocode *Sil* in table 1 of [1]) or “Massive sand” (lithocode *Sm* in table 1 of [1]) are identified during the visual classification. This could complicate the segmentation, particularly if a clustering approach is desired, where the model is not told explicitly how to interpret subclasses.
- (R2) *Insufficiency of images.*** The workload to label the images from the 9 non-public cores will likely be substantial. If the decision (due to e.g. lack of adequate time) is made to use only data from the 2 publicly available cores, there is a possibility that the total number of images may not be sufficient for adequate training and testing of a DL model.
- (R3) *Insufficient computing resources.*** The infrastructure of a conventional laptop or desktop may not be sufficient to run CNNs with this dataset.
- (R4) *CNN architecture does not achieve acceptable results.*** There is always a possibility that the model will not produce a valid or useful classification.

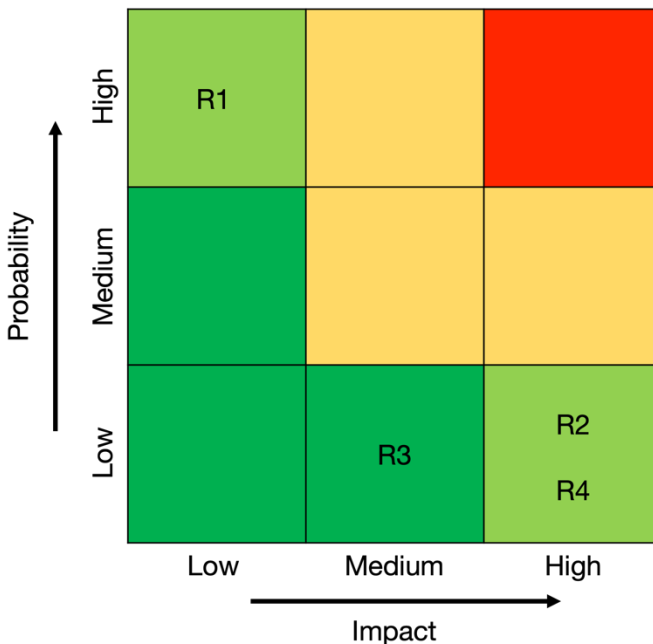
From the description of these risks and knowledge of the project goals and input dataset, the probability and mitigation strategies for each risk can also be identified:

- (M1)** R1 is quite likely to be encountered. However, it can be mitigated by adjusting (most likely increasing) the number of possible classes for the model to identify. Up to 14 classes were

reported in the visual classification of the core from site 2A [1]. An additional de-scoping strategy would be to eliminate the clustering approach and stick to supervised learning only.

- (M2) R2 is unlikely to materialise when one considers that the authors of [11] used a total of only 82 images and achieved good results (see section 5). Nevertheless, a mitigation strategy could be to use as many of the 11 cores as time allows, thereby increasing the total number of images from a minimum of 400 to somewhere below 1500.
- (M3) Considering the resources used by recent similar projects [7–11], and the relatively straightforward access to cloud-based GPU resources like Google-Colab, the probability of insufficient computing resources is low. However, depending on the power of these resources, a significant amount of time may be necessary to train the models anyway. Thus, additional mitigation strategies include simplifying the architecture and using pre-trained models.
- (M4) Again referencing the successes of past studies, it is possible to say at this stage that the likelihood of not finding a model that achieves acceptable litho-classification is low. However, it is expected that attempting different architectures and/or models with more or less parameters, as needed, will be part of the project, mitigating the risk of low-quality results.

Based on the risks and mitigation strategies, figure 5 shows a preliminary risk matrix for the project:

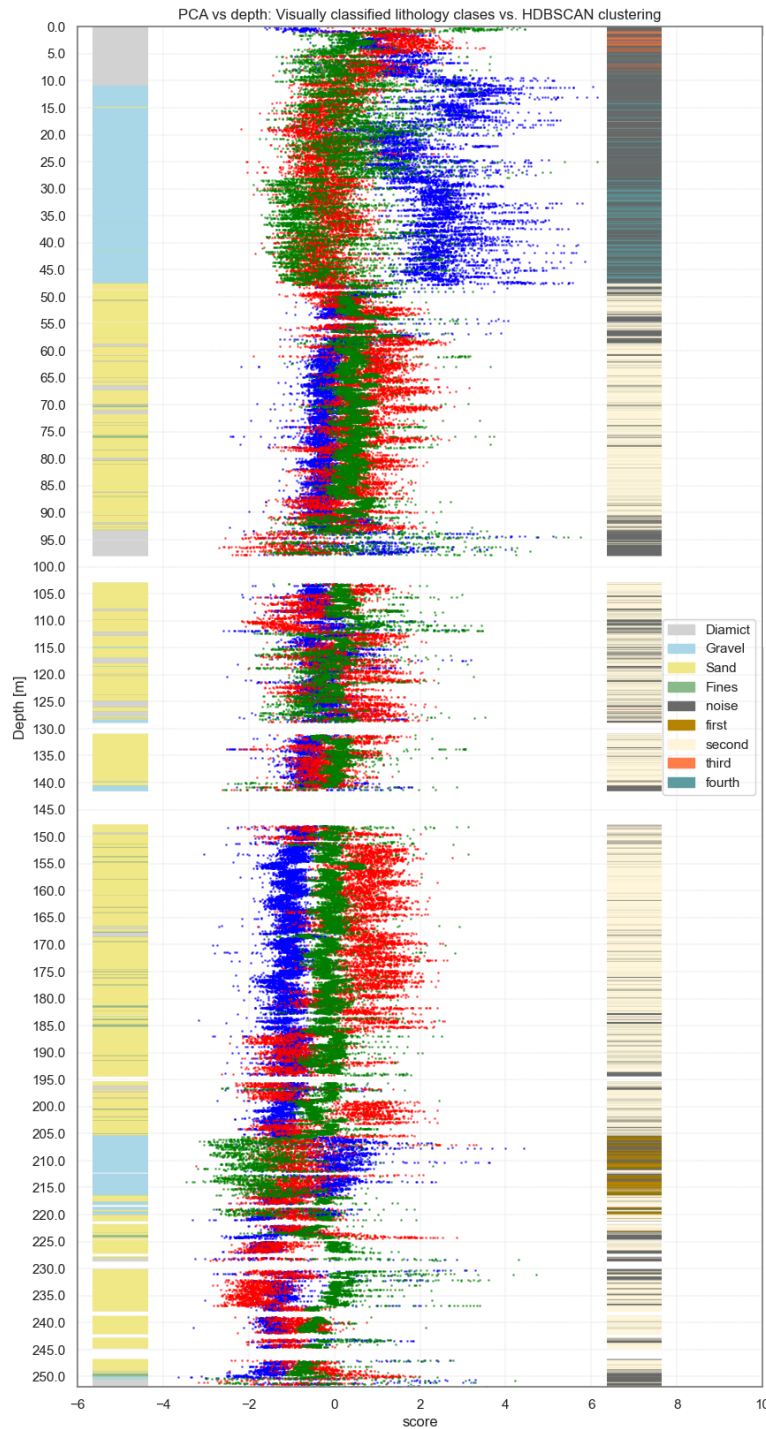


**Figure 5.** Risk matrix for the project. Green = low risk score; Light green = Medium-low score; Yellow = Medium risk score; Red = High risk score. All risks identified have a low to medium-low score.

The actual impact of these risks on the various aspects of the project, such as schedule, cost and quality, can be assessed within the context of the project environment. As the project is planned to be executed within the CAS program, time and resources are necessarily limited. Thus, the aim is to minimise these, even at the expense of some quality of the final model (quality here is understood to mean the capabilities of the algorithm to fully reach all of its objectives and achieve good automated classification). For this reason, all mitigation strategies have some aspect of de-scoping, such that the project will always be completed on schedule and without increasing resources too much, but it may lack some capabilities or features, such as clustering, or a dataset that uses all 11 cores.

## 10 Preliminary Studies

One of the main preliminary results related to this project was achieved during Module 3 of the CAS, which itself could be considered a developmental portion of this project and of [4]. By means of a PCA-driven dimensionality reduction of the MSCL log data, an HDBSCAN clustering algorithm was used to find natural clusters in the data. The parameters were tweaked such that 3 – 5 clusters were found. Figure 6 shows a comparison between the visual stratigraphic column of [1] and that found by the model.



**Figure 6.** Visual stratigraphic column (left) compared to the HDBSCAN column (right). The colours on the leftmost column indicate the visual classification of sediments at that depth (e.g. light blue represents gravel). The red, green and blue points in the centre represent the principal component scores vs. depth (see X-axis). The colours on the rightmost column indicate the clusters found by the HDBSCAN clustering model.

Large scale structures (e.g. the majority presence of sand) are clearly detected by the model. Furthermore, if one considers the points detected as noise to be their own cluster, two clusters were detected that seem to correspond to gravel sections between ~10 – 48 m and between ~205 – 220 m deep. Interestingly, these two gravel sections are actually distinguishable from each other even in the visual classification, if more than just the four major classes are considered.

## 11 Conclusions

The present CDR makes a clear argument for the usefulness of a DL algorithm that can perform automated litho-classification of sediment cores. Though similar studies have been published recently, only one is (indirectly) applicable to the Alpine foreland, and a generic core classification algorithm for use with scans of any type of core is still to be developed. Even if this project proves to only be applicable to the ROI and does not achieve generality, such a result would still be a valuable contribution to the overall efforts in the field to reach a generically applicable tool. The data, through collaboration with UniBe colleagues, is available, reliable, and already at a pre-processed stage. The data flow and methodology foundations are established and will be fine-tuned during the course of the project. Due to the state-of-the-art nature of this project, it is foreseeable that results could be published in specialised journals for further reference.

## Acknowledgements

I'd like to thank Sebastian Schaller for providing the information on the data (and eventually the data itself). Although our CDRs for Module 1 are independent, we will most likely work together in our final CAS projects to maximize the potential of the data and ML approaches.

## Statement

« Ich erkläre hiermit, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls die Arbeit als nicht erfüllt bewertet wird und dass die Universitätsleitung bzw. der Senat zum Entzug des aufgrund dieser Arbeit verliehenen Abschlusses bzw. Titels berechtigt ist. Für die Zwecke der Begutachtung und der Überprüfung der Einhaltung der Selbstständigkeitserklärung bzw. der Reglemente betreffend Plagiate erteile ich der Universität Bern das Recht, die dazu erforderlichen Personendaten zu bearbeiten und Nutzungshandlungen vorzunehmen, insbesondere die schriftliche Arbeit zu vervielfältigen und dauerhaft in einer Datenbank zu speichern sowie diese zur Überprüfung von Arbeiten Dritter zu verwenden oder hierzu zur Verfügung zu stellen. »

Date:

Signature(s):

15.12.2023

## References

- [1] S. Schaller, M. W. Buechi, B. Schuster, and F. S. Anselmetti (2023) Drilling into a deep buried valley (ICDP DOVE): a 252m long sediment succession from a glacial overdeepening in northwestern Switzerland. *Scientific Drilling* (2023), doi.org/10.5194/sd-32-27-2023
- [2] DOVE-Phase 1 Scientific Team, Anselmetti, F. S., Beraus, S., ... , Schaller, S., et al. (2023): Drilling Overdeepened Alpine Valleys (DOVE) – Operational Report of Phase 1, (ICDP Operational Report), Potsdam : GFZ German Research Centre for Geosciences, 70 <https://doi.org/10.48440/ICDP.5068.001>
- [3] <https://www.icdp-online.org/projects/by-continent/europe/dove-switzerland/public-data>
- [4] S. Schaller (2023), ADS CAS Conceptual Design Report: Core and wireline log based lithoclasification of unconsolidated sediments from drill cores and boreholes.
- [5] Lee, A.-S., Enters, D., Huang, J.-J.S., Liou, S. Y. H. & Zolitschka, B. (2022) An automatic sediment-facies classification approach using machine learning and feature engineering. *Nature Commun. Earth Environ.* 3, 1–9, doi.org/10.1038/s43247-022-00631-2
- [6] Xing, Y.; Yang, H.; Yu, W., (2023) An Approach for the Classification of Rock Types Using Machine Learning of Core and Log Data. *Sustainability* , 15, 8868. [doi.org/10.3390/su15118868](https://doi.org/10.3390/su15118868)
- [7] Alzubaidi, F., Mostaghimi, P., Swietojanski, P., Clark, S. R., & Armstrong, R. T. (2021). Automated lithology classification from drill core images using convolutional neural networks. *Journal of Petroleum Science and Engineering*, doi.org/10.1016/j.petrol.2020.107933.
- [8] Lauper B., Zimmerli G. N., Jaeggi D., Deplazes G., Wohlwend S., Rempfer J., Foubert A. (2021) Quantification of Lithological Heterogeneity Within Opalinus Clay: Toward a Uniform Subfacies Classification Scheme Using a Novel Automated Core Image Recognition Tool. *Frontiers in Earth Science*, 9. doi.org/10.3389/feart.2021.645596
- [9] Jacq, K. et al. Sedimentary structure discrimination with hyperspectral imaging in sediment cores. *Sci. Total Environ.* 817, 152018 (2022).
- [10] Fu D., Su C., Wang W., Yuan R. (2022) Deep learning-based lithology classification of drill core images. *PLoS ONE* 17(7). doi. org/10.1371/journal.pone.0270826
- [11] Di Martino, A., Carlini, G., Castellani, G. et al. (2023). Sediment core analysis using artificial intelligence. *Nature Sci. Rep.* 13, 20409. doi.org/10.1038/s41598-023-47546-2
- [12] Polonia, A., et al. "Recognizing mega tsunamis in Mediterranean deep sea sediments based on the massive deposits of the 365 CE Crete event." *Scientific Reports* 12.1 (2022): 5253.
- [13] Morcillo, G., Faria, S. H., and Kipfstuhl, S. (2020). Unravelling Antarctica's Past through the Stratigraphy of a Deep Ice Core: an Image-Analysis Study of the EPICA-DML Line-Scan Images. *Quat. Int.* 566-567, 6–15. doi:10.1016/j.quaint.2020.07.011
- [14] Apollo to Artemis: Drilling on the Moon (2021). <https://www.nasa.gov/centers-and-facilities/kennedy/apollo-to-artemis-drilling-on-the-moon/>
- [15] Deep drilling for Mars (2023). [https://www.esa.int/ESA\\_Multimedia/Images/2023/03/Deep\\_drilling\\_for\\_Mars](https://www.esa.int/ESA_Multimedia/Images/2023/03/Deep_drilling_for_Mars)
- [16] Becerra, P., I.B. Smith, et al. (2021), Past, Present and Future of Mars Polar Science: Outcomes and Outlook from the 7th International Conference on Mars Polar Science and Exploration, *The Planetary Science Journal*, 2:209, doi.org/10.3847/PSJ/ac19a5
- [17] Van Rossum, G. (2022). os—Miscellaneous operating system interfaces. <https://docs.python.org/3/library/os.html>
- [18] The Pandas Development Team. (2022). pandas-dev/pandas: Pandas (v1.5.1). Zenodo. <https://doi.org/10.5281/ZENODO.3509134>
- [19] Harris C.R., et al. (2020) "Array programming with NumPy," *Nature*, vol. 585, no. 7825, Art. no. 7825, doi: 10.1038/s41586-020-2649-2.
- [20] P. Virtanen et al (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, *Nat. Methods*, vol. 17, pp. 261–272. doi: 10.1038/s41592-019-0686-2.



- [21] S. Seabold and J. Perktold (2010) “statsmodels: Econometric and statistical modeling with python,” in 9<sup>th</sup> *Python in Science Conference*. <https://www.statsmodels.org/stable/index.html>
- [22] F. Pedregosa et al. (2011) “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. Oct, pp. 2825–2830, <https://scikit-learn.org/stable/about.html>
- [23] M. Abadi, A. Agarwal, et al. (2015) TensorFlow: Large-scale machine learning on heterogeneous systems,. Software available from tensorflow.org
- [24] S. van der Walt, J. L. Schönberger, et al. (2014) scikit-image: Image processing in Python. *PeerJ* 2:e453. <https://doi.org/10.7717/peerj.453>
- [25] Clark, J. et al. (2010). PIL : Python Imaging Library, <https://pypi.org/project/Pillow/>
- [26] OpenCV. (2015). Open Source Computer Vision Library. <https://opencv.org/>
- [27] Caswell, T. A., Lee, A., et al. (2022). matplotlib/matplotlib: REL: v3.6.1 (v3.6.1). Zenodo. <https://doi.org/10.5281/ZENODO.592536>
- [28] Y. Ikeda, “mpltern: Ternary plots with Matplotlib.” Zenodo, 2023. doi: <https://doi.org/10.5281/zenodo.3528354>.
- [29] Waskom, M. (2021). seaborn: Statistical data visualization. *Journal of Open Source Software* , 6 (60), 3021. <https://doi.org/10.21105/joss.03021>
- [30] Plotly Technologies Inc. (2015). Collaborative data science. Plotly Technologies Inc. <https://plot.ly>
- [31] Ronneberger, O., Fischer, P. & Brox, T. (2015) U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Interventio*. doi.org/10.1007/978-3-319-24574-4\_28.
- [32] Tan, M. & Le, Q. V. (2020). EfficientNet: Rethinking model scaling for convolutional neural networks. <https://doi.org/10.48550/arXiv.1905.11946>.
- [33] He, K., Zhang, X., Ren, S. & Sun, J. (2020). Deep residual learning for image recognition. doi.org/10.48550/arXiv.1512.03385.
- [34] Russakovsky, O. et al. (2015). ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 211–252.