ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Master Thesis

Deep Feature Learning for Digital Elevation Models with Auto-encoders

Author:
Reto Spannagel

Supervision:
Prof. Dr. Lorenz Hurni
Dr. Magnus Heitzler
Sidi Wu
Marianna Farmakis-Serebryakova

Institute of Cartography and Geoinformation
ETH Zurich

Submission date:
January 31, 2022

Acknowledgements

I would like to thank Prof. Dr. Lorenz Hurni and the Institute of Cartography and Geoinformation (IKG) at the Swiss Federal Institute of Technology Zurich (ETH) for making this master thesis possible and providing the best possible working environment during the pandemic.

I am very grateful to my supervisors Dr. Magnus Heitzler, Sidi Wu and Marianna Farmakis-Serebryakova for their excellent advice, support and feedback whenever needed - both at the Institute and via Zoom meetings. I would also like to thank Raimund Schnürer and Xue Xia for welcoming and supporting me at my second workplace at their office.

Last but not least, I would like to thank my family and my flatmate for supporting me at any time and for reading through my thesis.

Abstract

This master thesis deals with the automatic classification of landscape types, such as mountains, valleys, or hills, in an unsupervised manner based on a Digital Elevation Model (DEM). Auto-encoder neural networks compress the original input into a feature vector representation in the encoder and recreate the input with the decoder. In the process, the feature vector representation is trained to contain characteristic information about the landscape terrain. Different architectures and model parameters are tested to optimize the feature extraction process. The features are then used to subdivide the DEM into different landscape classes based on their common characteristics using a clustering algorithm. The results, which are compared with the existing *Landscape Typology of Switzerland*, demonstrate that auto-encoders can be used to extract reasonable classes for the study area of Switzerland, although not all types of manually distinguishable landscape classes can be found, and optimizing the results is restricted by computational limitations.

Keywords: landscape classification, auto-encoders, digital elevation model, clustering

Contents

List of Figures

List of Tables

1 Introduction

In this thesis, auto-encoders are used to classify landscapes. The general idea is to generate different classes of landscapes based on their elevation structure. For this, digital elevation models (DEM) are used as input data. The classification is conducted using auto-encoder neural networks. The main objective is to investigate and employ suitable methods to train the network and obtain meaningful features, which relate to characteristics of a given landscape. These features are then used to segment the study area into classes of landscapes.

Auto-encoder models, that have been used in the past in various fields for classification purposes, are adapted in this thesis to the purpose of landscape classification. An adequate auto-encoder model should be able to extract useful features under reasonable computation cost.

In order to obtain landscape classes, clustering algorithms are used. Landscapes could potentially be classified based on physical, geological, hydrological, biological or cultural characteristics. However, the scope of this thesis is restricted to landscape classification based on physical features, which can be derived from DEMs. Potential landscape types that could be derived from altitude information include mountains, hills, valleys or plains. To obtain such classes, the feature space representation should contain some characteristic features for different landscapes other than pure altitude information.

A DEM of Switzerland will be used as input to obtain a classification of the landscapes of Switzerland. While there are many potential subdivisions that could be considered reasonable, the success will be measured based on an existing manual classification.

2 Literature Review

Currently, no literature exists specifically on unsupervised classification of landscapes based on DEM using auto-encoders, however there is a wide range of literature on related topics.

Extensive research on unsupervised image classification and clustering based on auto-encoders exists primarily on the MNIST dataset (Guo et al., 2017; Dong et al., 2020). Semantic segmentation of large-scale images has been used for various applications such as classification of rock types (Karimpouli and Tahmasebi, 2019), polarimetric SAR imagery (Mohammadimanesh et al., 2019), binary classification of seamounts based on topographic features (Valentine et al., 2013) and annotation of seafloor imagery (Yamada et al., 2021). Badrinarayanan et al. (2015) developped an auto-encoder-based approach for pixelwise image segmentation.

Ehsani and Quiel (2009) classified landscapes using both satellite imagery and DEMs. The feature space representation of landscape elements is also studied by relating the resulting clustered classes to different attributes such as slope, maximum and minimum curvature. Du et al. (2019) classified landforms in a supervised manner by relating visual features from shaded reliefs and physical features from DEM and slope data. Three fully connected convolutional layers and a softmax classifier are used to predict the classes for the sample data. Using the combination of input data, they achieved recognition accuracies of 83 % to 98 %, depending on the landform class, significantly higher than with only one type of input data. Similarly, Li et al. (2020) employed a U-Net network using DEM and image data as well, achieving an accuracy of 87 %. Additionally, they showed that the CNN-based approach outperformed an alternative random forest approach. Drăguţ and Blaschke (2006) explored the classification of Mars landforms using geographic information systems (GIS), object-based image analysis and digital terrain models (DTM) using a-priori

rules bases on expert knowledge for clustering. Bue and Stepinski (2006) used the Self-Organizing Map (SOM) neural network to find four different characteristic Martian landforms (highlands, craters, lowlands and high relief) by grouping together 20 different classes. For the input data, six different attributes including elevation and slope derived from standard DTMs and *flooded* DTMs (representing enclosed depressions) are used. Stepinski et al. (2006) segmented Mars landscapes using slope, curvature and flood attributes in an unsupervised manner. In a second step, the segments were classified in a supervised manner to match six pre-determined classes. The most accurate learning algorithm was a Support Vector Machine (SVM), with 85 % accuracy. Other non-CNN-based methods include classification by fuzzy rules (Drăguţ and Blaschke, 2006) or the random forest method (Zhao et al., 2017).

K-means is considered the main approach for remote sensing clustering applications (Bue and Stepinski, 2006). Nousi and Tefas (2020) propose several adjustments to the k-means algorithm to improve the the intra-cluster compactness and the inter-cluster separability. Li et al. (2017) introduce a technique called Discriminatively Boosted Clustering (DBC) to improve the separability of clusters. Gansbeke et al. (2020) propose decoupling the process of representation learning and clustering in a two-step approach, although there are also methods which combine the feature learning process with a standard k-means clustering algorithm (Caron et al., 2019; Dong et al., 2020).

3 Method

3.1 Data

The digital elevation model (DEM) used in this thesis encompasses the entirety of Switzerland and surroundings with a resolution of 40x40 m per pixel. The topography of Switzerland can generally be described as consisting of three main regions. The *Mittelland* (*Swiss Plateau* or *Central Plateau*) covers approximately 30 % of the area of Switzerland. Although some parts are rather flat, the majority of its extent is characterized by hilly landscapes. To the south, it is bordered by the Alps, which cover about 60 % of Switzerland. This area consists of larger mountainous areas as well as many valleys, which cut through its extent. Furthermore, a strip along the Alps, bordering the Mittelland, is commonly referred to as the Prealps, which is a transition zone between the mountainous areas and lower hilly areas. To the northwest of the Mittelland lies the Jura (10 % of Switzerland), which is another mountainous area although with lower altitudes than the Alps (EDA, 2021).

For the validation of the landscape classification, the *Landschaftstypologie Schweiz* (Landscape Typology of Switzerland), developed by the Federal Office for Spatial Development (ARE), the Federal Office for the Environment (BAFU) and the Federal Statistical Office (BFS) is used (ARE, 2021). It divides the area of Switzerland into 38 different landform types based on topographic, geological, demographic and ecological features.

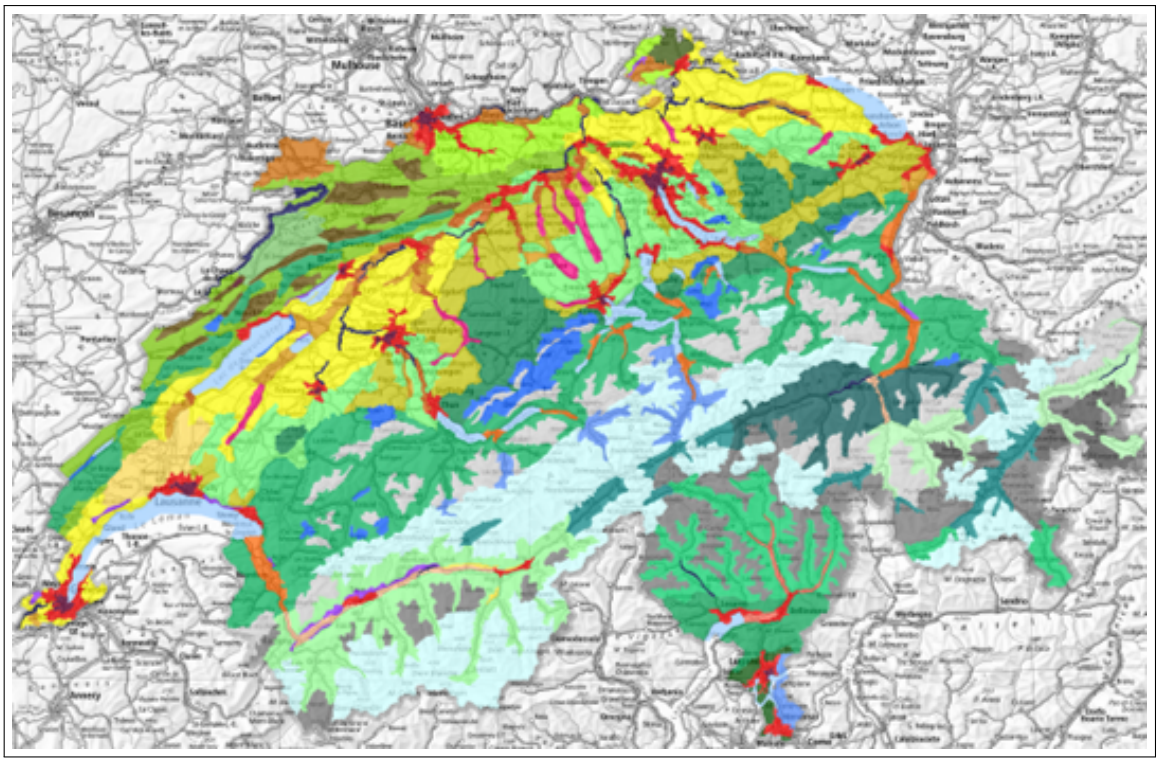

Figure 3.1: Original Landscape typology provided by ARE, BAFU and BFS (ARE, 2021)

3.2 Autoencoders and Feature Extraction

Auto-encoders are a subclass of artificial neural networks (ANNs) that are different from other types of artificial neural networks by its desire to produce an output that is identical to the input (Valentine and Trampert, 2012). It consists of an encoder followed by a decoder. The original data x are mapped to a latent representation h of lower dimensionality, which can be expressed as $h = f_\phi(x)$. This process usually includes non-linear transformations. The decoder reconstructs x_r from the latent representation, which can be expressed as $x_r = g_\theta(h)$ (Yamada et al., 2021). ϕ and θ in this case are the encoder and decoder parameters. Both the encoder and decoder may include one or several layers of neurons. The depth of a network is defined by the number of such layers. The reconstruction x_r should be as similar as possible to the input x, given the stored information of the latent representation h. Such an objective function can be formulated as shown in Equation 3.1, given n samples (Yamada et al., 2021):

$$\min_{\phi,\theta} L_{rec} = \min \frac{1}{n} \sum_{i=1}^{n} ||x_i - x_n||^2 \tag{3.1}$$

By downsampling the result at each layer, the next convolutional layer includes information for each pixel from a larger context. Since the classification of a single square in the DEM as part of a landscape requires information about a large number of surrounding squares, it is crucial to include such steps in a network.

In order to learn meaningful features, the input data needs to be compressed, forcing the network to keep only the most important information. Ideally, in an encoding process, the auto-encoder finds common characteristics of the input data (Valentine and Trampert, 2012). These common characteristics can be interpreted in this case as characteristics of a landscape type. By minimizing the loss function according to Equation 3.1, the network can learn weights (included in the parameters ϕ and θ), which filter the useful information from negligible information. Those weights are stored in weight matrices W, which exist between each pair of adjacent layers (Valentine and Trampert, 2012) (Nousi and Tefas, 2020). Generally, the latent representation h should be small, since relevant features can be sufficiently represented by a feature vector of compact size and for the purpose of clustering it would be

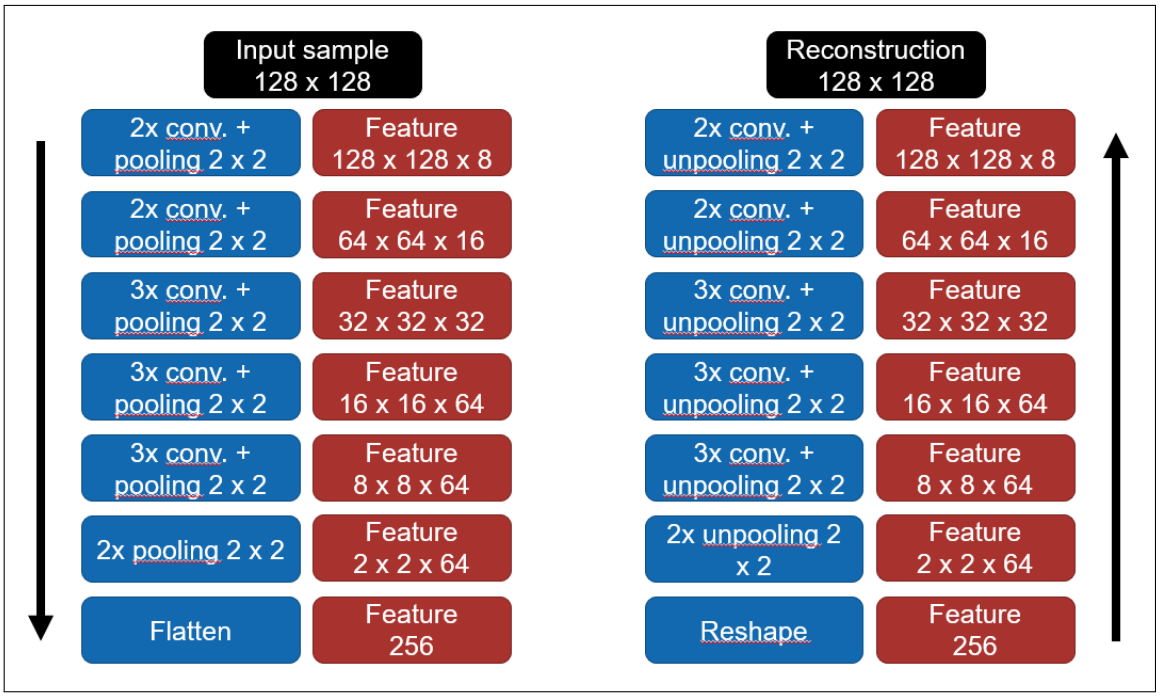

Figure 3.2: Schematic representation of the modified SegNet architecture (Model B)

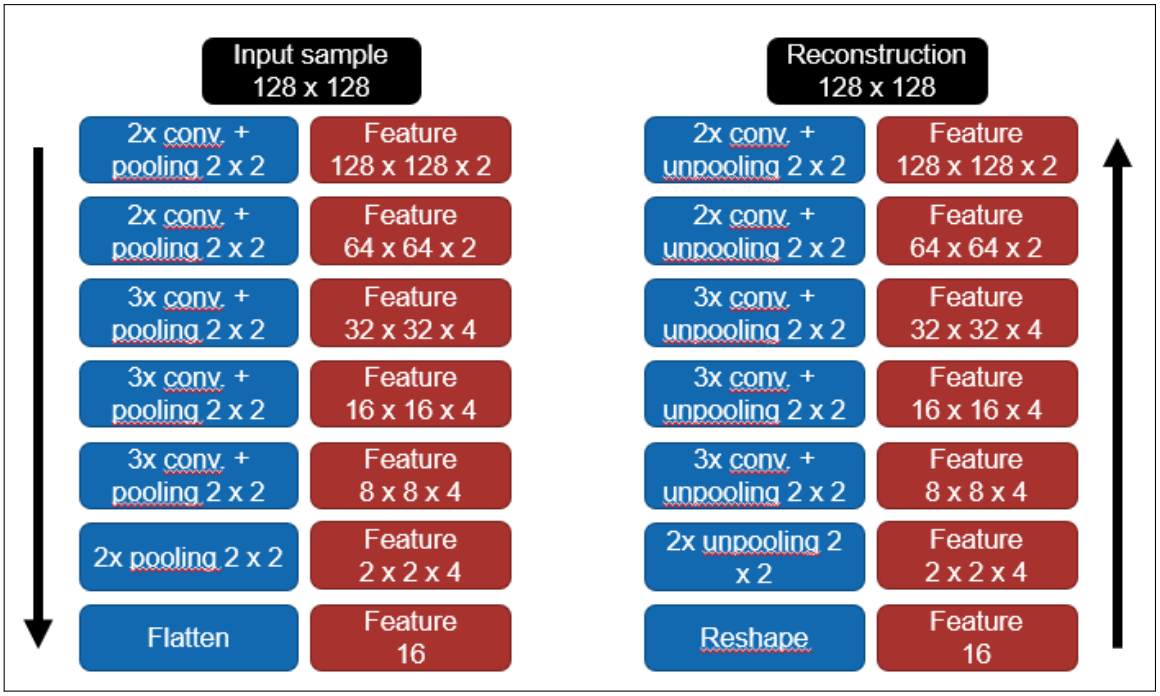

Figure 3.3: Schematic representation of the modified SegNet architecture (Model D)

disadvantageous to use an unnecessarily large amount of data. Useful information would be blurred among less useful information. At the same time, excessive compression would lead to the loss of information that is necessary to reconstruct x_r. Therefore, the optimal size of the latent representation and the suitable architecture to extract such a representation needs to be determined experimentally, as no general rule exists for all possible datasets (Yamada et al., 2021; Mai Ngoc and Hwang, 2020).

At early stages, a basic autoencoder architecture has been tried, which did not provide satisfactory results with regards to the reconstruction of the DEM. Yamada et al. (2021) and Catani (2020) proposed using the AlexNet architecture for feature learning with the purposes of annotating seafloor imagery and detecting landslides from optical images.

The final autoencoder architecture used in this thesis, which provided the most promising results, is the SegNet architecture, developed by Badrinarayanan et al. (2015). It is a deep convolutional neural network architecture, based on the VGG16 network, designed for semantic pixel-wise segmentation of RGB-images. Two of the used variations of this network are shown in Figures 3.2 and 3.3. The original SegNet architecture has a symmetrical structure regarding the encoder and the decoder networks and features 13 convolutional layers on each side. At each of the five levels of the network, two or three convolutional layers are applied, followed by batch-normalization, a rectified-linear unit (ReLU) and a max-pooling layer with a 2x2 window to downsample the input data by a factor of 2. The batch-normalization layer is used to standardize the inputs, which accelerates training. The ReLU layer introduces nonlinearity, which is necessary as during the entire convolutional process, only linear operations such as multiplications and summations are applied. The ReLU activation function simply changes all negative values to zero, as shown in Equation 3.2 (Karimpouli and Tahmasebi, 2019):

$$f(x) = max(0, x) \tag{3.2}$$

The max-pooling layer summarizes the data in the window by taking the local maximum. This process is repeated throughout the entire sample using a sliding window.

Badrinarayanan et al. (2015) have shown that the SegNet network outperforms similar networks such as VGG16, U-net or DeconvNet regarding computing memory consumption. The advantage comes from storing only the locations of the max-pooling features rather than using fully connected layers. In the decoder section, the feature maps can be upsampled again using the stored max-pooling indices.

Karimpouli and Tahmasebi (2019) have shown that extending the SegNet network with regards to its depth is possible and necessary for more complex input data, as doing so produced more reliable results. Therefore, the original SegNet architecture is adapted for the purposes of this thesis by adding two additional max-pooling levels without further convolutional layers, resulting in seven max-pooling layers overall on the encoder side. Correspondingly, unpooling layers on the decoder side are added as well. Using this system, an input sample with 128x128 (2^7x2^7) pixels is downsampled to a resolution of 2x2 pixels. At this stage, the feature map has 32 channels, up from the original single-channel greyscale image. In the last step of the encoder, the 2x2x32 feature map is flattened to a feature vector with a length of 128, which is necessary to forward the data to the clustering algorithm.

The model is implemented using Anaconda in Python 3.6.13 with Keras 2.2.4, which uses TensorFlow 1.15.0. All the training and testing is conducted on the GPU (NVIDIA GeForce GTX 1080 Ti) using the CUDA Toolkit 10.0.130. For the optimizer, the Adam algorithm is used with the default learning rate of 0.001. For the loss function, which quantifies the error from the predictions with regards to the labels, a mean squared error is used. For the *metrics* option, *accuracy* is used, which simply calculates how often the predictions equal the labels. In the training stage, the network uses a sample of 128 images of the size 128 x 128 pixels, while 32 images of the same size are used for validation. Both sets are sampled from the input DEM using a set of fixed numbers which are originally generated randomly from a fixed seed to avoid bias from hand-picking samples. The model is trained with 1000 epochs, using the same set of images. The batch size is kept at 16 samples. Therefore, at each epoch, 8 batches of the same samples are used.

Variations of the model were tested on a study area of the size 50 km x 50 km in eastern Switzerland. The area contains all four landscape classes as described in

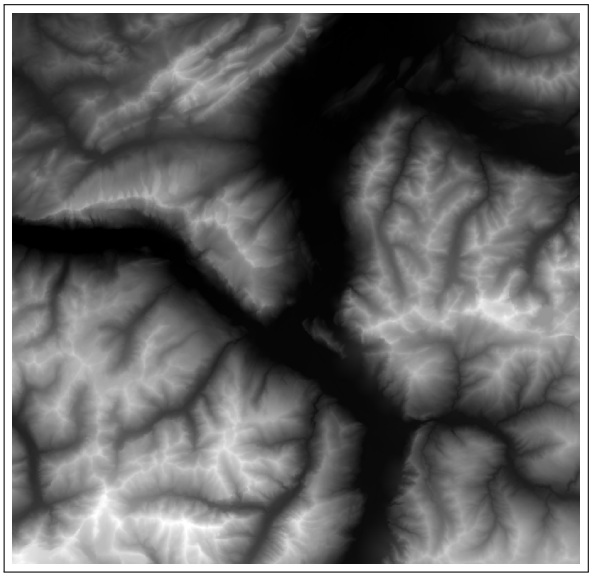

Figure 3.4: Original DEM of the study area

Section 3.4, which makes it a useful test site for different settings, as testing on the complete data would be computationally expensive.

In a first step, the variations are tested with regards to their ability to reconstruct the DEM (see Figures 3.4 and 3.5). As explained above, there is no need for a perfect reconstruction, however, the original structure of DEM should be recognizable. Specifically, characteristic structures such as mountains, hills, valleys or flat plains should be preserved. If that is not the case, the learned features do not accurately represent the input data and therefore will not be useful for classification. In a second step, the features are used for clustering and the results are reviewed, in order to improve the model based on a trial-and-error approach. First, some of the basic parameters, as shown in Table 3.1, are fixed. Afterwards, some structural settings such as number of layers and size of the feature vector were tested. The variations, which are used to produce the results shown in this thesis (Models A to D) are further explained in Section 4.1.

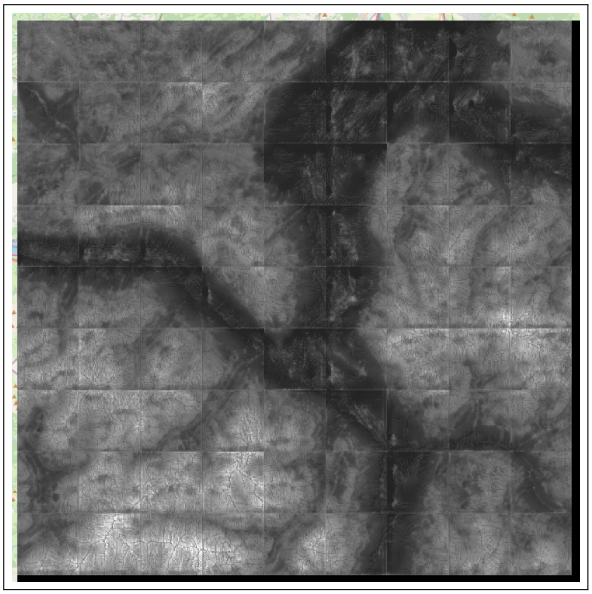

Figure 3.5: Reconstructed DEM using Model B

Table 3.1: Settings for Model B

Parameter	Value
Input sample size	128 x 128 px
Kernel size	3 x 3
Batch size	16
Training images	128
Validation images	32
Epochs	1000

3.3 Clustering

Clustering is a task in data analysis whereby the data are divided into a number of groups based on common or similar features. Since the clusters do not have to be pre-defined, it is considered a method within the field of unsupervised learning. The advantage to classification, which is a task of supervised learning, lies within the possibility to find similar features without the need to provide labeled training data. Although such training data might be useful to match the resulting classes with what we would consider reasonable landscape classes, this thesis focuses on providing a method of unsupervised learning. Technically, by using clustering, a segmentation technique is used in this thesis rather than classification.

There are generally two types of clustering algorithms, being hierarchical and partitional clustering. Hierarchical clustering algorithms group together data in a way which can be represented using a dendrogram. The bottom-up (agglomerative) approach pairs most similar features first and gradually merges all clusters. The top-down (divisive) approach gradually divides all data points into smaller clusters. Well-known hierarchical clustering algorithms are single-linkage and complete-linkage. Partitional clustering algorithms on the other side finds all the clusters simultaneously. K-means is generally considered the most popular partitional clustering algorithm (Jain, 2010; Nousi and Tefas, 2020).

The k-means algorithm uses a pre-defined number of clusters K. It aims to partition the N observations within a set of observations $X = (x_1, x_2, ..., x_N)$, with $K < N$, into a set of clusters $C = (c_1, c_2, ..., c_K)$ in such a way that the observations within each cluster are as similar to each other as possible. This is done by minimizing the variance, i.e. the squared error between the mean of data points in each cluster and the points within the cluster. The objective function to be minimized can therefore be written according to Equation 3.3 (Jain, 2010):

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} ||x_i - \mu_k||^2 \tag{3.3}$$

The k-means algorithm follows three main steps. First, an initial partition is selected, where the cluster centers can be assigned randomly. In the second step, each data point is assigned to the closest cluster center. In the first step, the new cluster centers are re-calculated based on the points assigned in step two. Steps two and three are repeated until the algorithm converges, i.e. the cluster membership remains stable (Jain, 2010). A known issue is that the k-means algorithm converges to local minima, depending on the initial partition.

3.4 Evaluation

Since the goal is to retrieve reasonable landscape classes, the results cannot easily be verified (Drăguţ and Blaschke, 2006). A possible method is to compare the computed landscape classes with existing subdivisions based on expert knowledge. For

this, the Landscape Typology of Switzerland is used, as described in Section 3.1. It should however be noted, that the results cannot be expected to match a deliberate classification to a large extent, as the reference classification is not used in the training process like it could have been in a supervised learning approach. Nonetheless, a comparison could give a general idea of the performance of the model.

Since the Landscape Typology of Switzerland contains 38 different classes, partially based on data not related to altitude, a reduced model is used for comparison purposes. It contains four main classes, which could be described as *Plains* (A, blue), *Hills* (B, purple), *Lower Alps* (C, green) and *Higher Alps* (D, turquoise), as shown in Figure 3.6. Since the reference model only covers the area within the borders of Switzerland, the results have also been cropped to the national border.

Additionally, the results can be evaluated visually, by draping them over the input DEM in a GIS software and comparing the classes and their borders with the terrain features (Drăguţ and Blaschke, 2006).

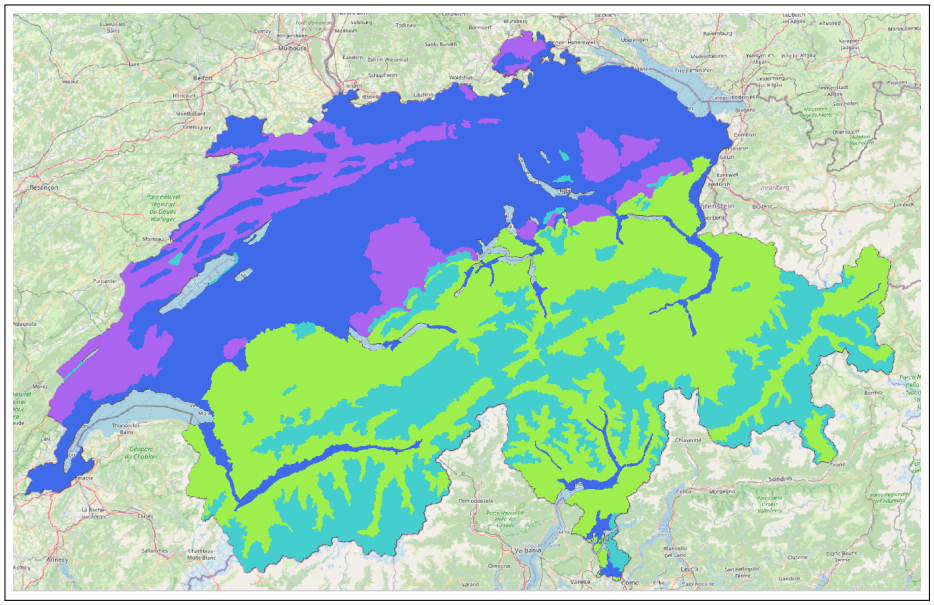

Figure 3.6: Four main landscape classes generated from the Landscape Typology of Switzerland

4 Results

4.1 Results for Study Area

Different model settings were tested on a study area of the size 50 km x 50 km in eastern Switzerland. The area contains all four landscape classes as described in Section 3.4, which makes it a useful test site for different settings, as testing on the complete data would be computationally expensive.

Figure 4.1 shows a visualization of the feature vector representations using a variation of the model described in Section 3.2, where each training image of size 128 x 128 pixels is downsampled to a single pixel using additional convolutional and max-pooling layers (Model A). The first three out of 128 channels are coded in RGB-colors. Figure 4.2 shows the subsequent clustering result from Model A. Although the result shows that such a model could differentiate plains from mountains, four of the shown classes should belong to the same class, reducing the number of distinct, semantically meaningful, classes to only two.

In Figure 4.3, the number of downsampling and convolutional layers is reduced to a model where the lowest layers contains features of size 2 x 2 x 128 (Model B). In this model, the classes are distributed in a more useful way. The classes could be interpreted as plains (blue), hills (purple), lower Alps (red and yellow) and higher Alps (green). However, in some parts such as the bottom-right corner, the classification seems to show some undesirable anomalies. If the model is trained with 5000 instead of 1000 epochs (Model C), these anomalies disappear (see Figure 4.4), however the *hills* class disappears, possibly due to overfitting.

A similar result (Model D), although much blurrier, can still be achieved by drastically reducing the number of feature channels from 128 to 16 (see Figure 4.5).

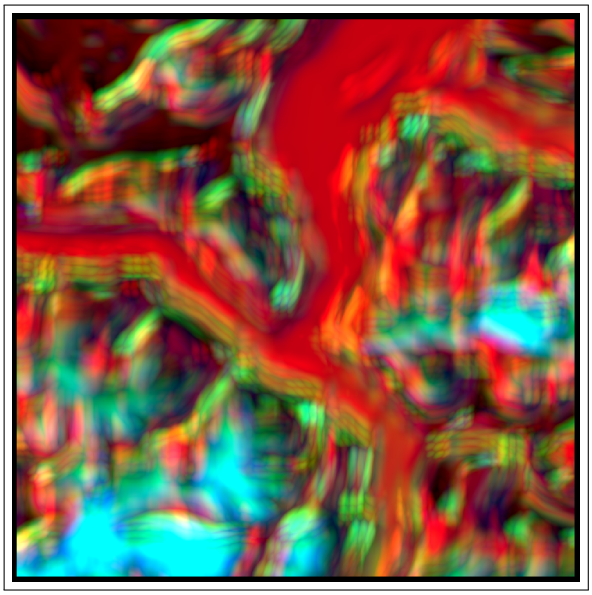

Figure 4.1: Visualization of the features (Model A)

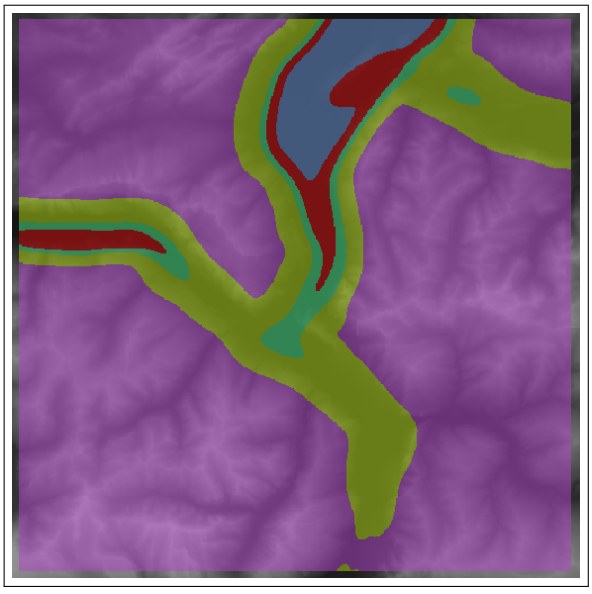

Figure 4.2: Clustering results for Model A

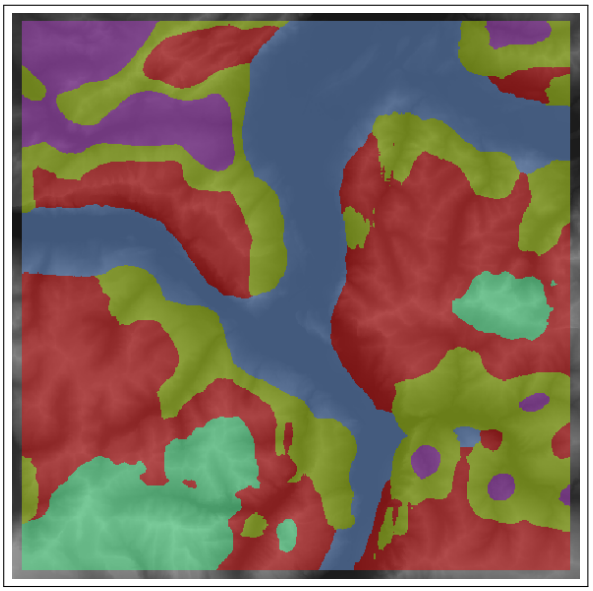

Figure 4.3: Clustering results for Model B

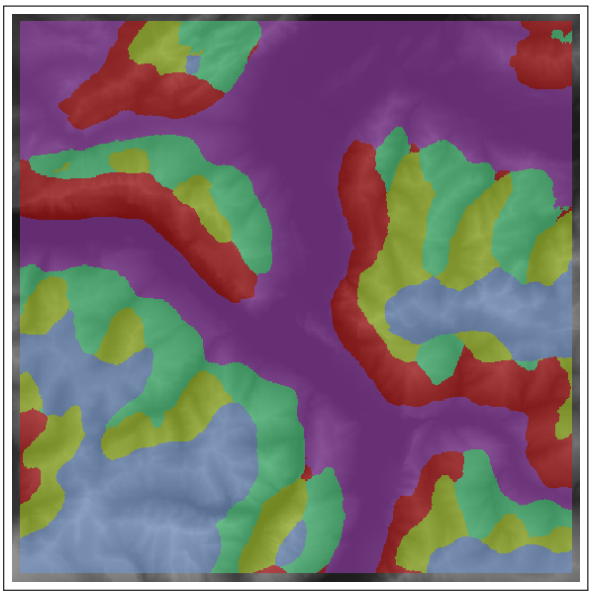

Figure 4.4: Clustering results for Model C

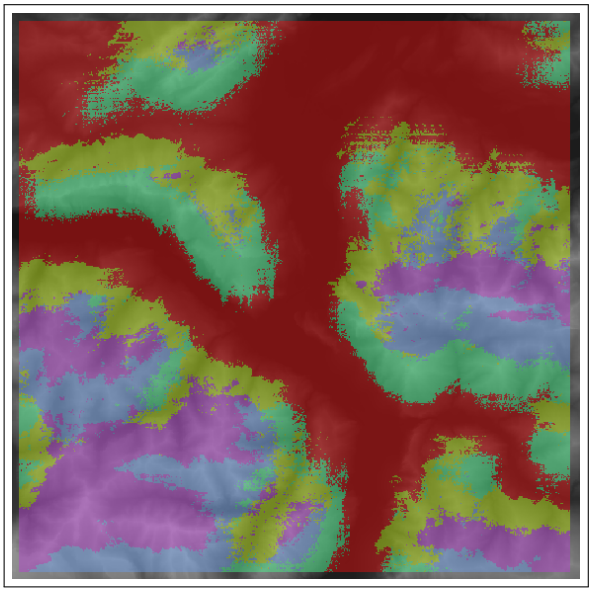

Figure 4.5: Clustering results for Model D

4.2 Results for Complete Data

Although Model B showed the most promising results, it proved to be impractical for the complete dataset, as the 128-channel feature representation of the dataset had a size of roughly 56 GB, which overloaded the clustering algorithm. Using Model B, a result could therefore not be generated. Subsequently, Model D was chosen to research if a reasonable result could be generated for the complete dataset, since Model D only uses a feature vector of size 16 rather than 128 (see also Figures 3.2 and 3.3). As expected, reducing the input data size for the clustering algorithm considerably, allowed generating a result for the complete dataset. Figure 4.6 therefore shows the results using Model D. Initially, the clustering is conducted using $K = 10$ clustering classes. This allows combining similar classes to four main classes resembling the modified reference classes from the Landscape Typology of Switzerland. The shown classes are *Plains* (1, pink), *Hills* (2, yellow), *Lower Alps* (3, green) and *Higher Alps* (4, red).

The results are then compared with the reference classes, where the corresponding classes are *Plains* (A), *Hills* (B), *Lower Alps* (C) and *Higher Alps* (D). Table 4.1 shows the percentage of pixels for each class from Model B that lie within each class

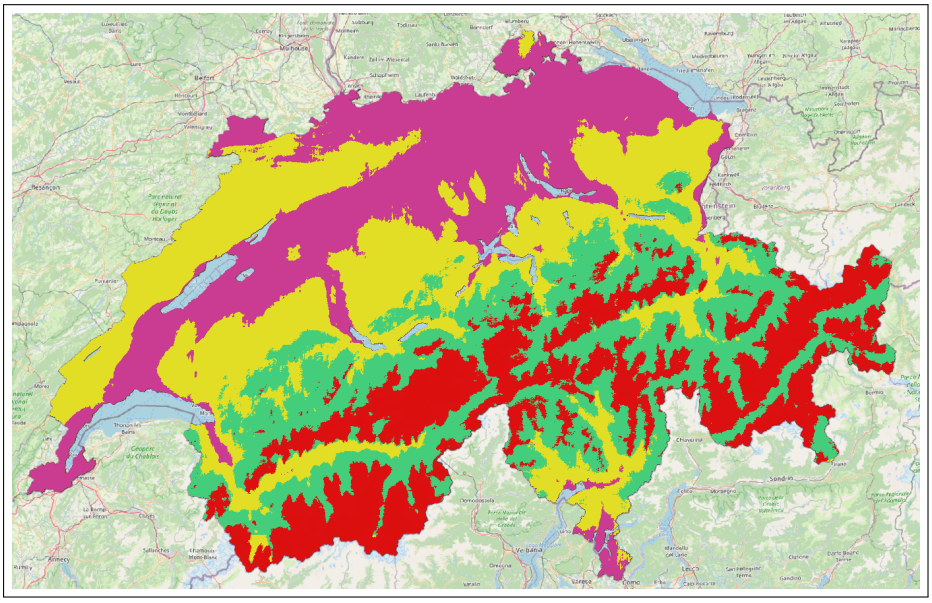

Figure 4.6: Clustering results from Model D after manually joining similar classes

of the reference classes, whereas Table 4.2 shows the comparison in the opposite direction.

It can be seen that Class 1 lies almost completely within reference Class A, however only two-thirds of Class A are within Class 1, indicating that Class 1 covers more area. Indeed, approximately one third of Class 2 also lies within reference Class A. This should not be seen as a problem since the definition of landscape classes can be flexible. Visually, the borders of Class 1 seem to be easily justifiable. The same also applies to Class 4. Class 2 (and to a lesser extent Class 3) is somewhat more problematic since it covers both lower mountain ranges such as the Jura and relatively flat Alpine valleys such as the Rhone valley, which should be found in different classes. This is also reflected in the numeric comparison, as Class 2 is evenly scattered through the reference classes A, B and C.

Table 4.1: Percentage of clustering classes within reference classes

Class	Class A	Class B	Class C	Class D
Class 1	89.1 %	8.5 %	1.4 %	1.0 %
Class 2	33.7 %	32.6 %	30.3 %	3.4 %
Class 3	0.2 %	0.3 %	75.3 %	24.2 %
Class 4	0.0 %	10.0 %	31.3 %	68.7 %

Table 4.2: Percentage of reference classes within clustering classes

Class	Class 1	Class 2	Class 3	Class 4
Class A	67.8 %	32.1 %	0.1 %	0.0 %
Class B	17.2 %	82.2 %	0.6 %	0.0 %
Class C	1.0 %	26.8 %	53.4 %	18.8 %
Class D	1.1 %	4.8 %	27.6 %	66.4 %

5 Discussion

The results for the complete dataset, using Model D show a classification that can be considered reasonable. When combining some of the produced classes, each of the resulting class can be given a plausible name. Still, some post-processing is necessary to produce well-rounded classes. Additionally, it should be noted that while the class borders do not directly correspond to contour lines, the classes are still heavily correlated with the altitude. Figure 5.1 shows a DEM of Switzerland after assigning four classes purely based on altitude intervals, for comparison. Finding distinct classes with differing attributes other than altitude seems to be challenging. With regards to that, Model B seems to show a more promising result, at least for a smaller area. Two of the classes (blue, and purple) seem to represent areas with relatively low slope values but different levels of altitude, while two more classes (yellow and red) cover areas with similar altitudes as purple but with higher slope values (see Figure 4.3).

A drawback of the k-means clustering in unsupervised methods is that the learned clusters do not necessarily align with semantically useful classes. This is especially the case when the class sizes are imbalanced, which is a typical problem with k-means clustering (Gansbeke et al., 2020).

Regarding the evaluation of the results, as explained in Section 3.4, the quality cannot easily be quantified using a reference classification, since the model is never trained to match a specific arbitrary manual classification. However, the comparison with the Landscape Typology of Switzerland is the only available quantitative evaluation metric and it can give a general idea of the ability of the model to recreate meaningful landscape classes.

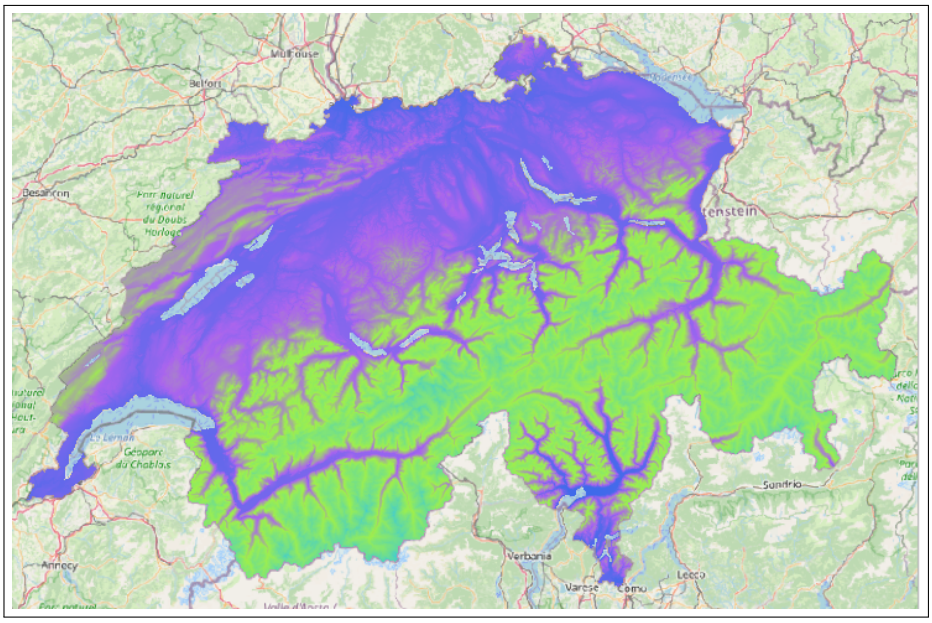

Figure 5.1: Classes purely based on altitude

6 Conclusions and Outlook

This thesis shows that it is possible to use auto-encoders to obtain some landscape classes from a DEM. Bue and Stepinski (2006) have already shown that it is possible to obtain classes that are either characterized by high or low elevation, or high or low slope, which can be confirmed in this thesis. It also shows some of the limitations with regards to the complexity of the obtainable classes. Retrieving landscape classes based on more sophisticated patterns remains challenging. It would be preferable to be able to differentiate landscapes such as the Jura mountains and the Pre-Alps based on their distinct structures and patterns. Possibly, a supervised learning approach could help forcing the model to use other classifiers. Furthermore, adding other types of input data not related to altitude such as aerial photographs or vegetation data could be helpful.

A possible improvement to the overall workflow lies in applying a method where the two main steps considered in this thesis, the feature learning and clustering steps, are jointly executed so that the loss function could include the clustering performance, as proposed by Li et al. (2018) or Yamada et al. (2021). Additionally, the k-means algorithm could be extended using the proposals by Nousi and Tefas (2020) or other types of clustering algorithms could be used such as the SCAN-algorithm proposed by Gansbeke et al. (2020), or Principal Component Analysis (PCA).

This thesis also outlines the tensions that arise between the necessity of using deeper and more complex models to obtain better results and the challenges that lies within the large amount of data to process. The results showed that reducing the model complexity may be necessary but could limit the classification performance. It was not possible to use the preferred model for the entire dataset. As a consequence, better results were produced for a smaller study area. Potentially, this issue could be addressed in the future by using input data with a smaller resolution or using

a more computationally efficient clustering algorithm. However it remains unclear, if using the preferred model would substantially improve the result for the entire dataset. Additionally, using input data with lower resolution would lose data which could be significant to characterize complex landscape types.

Bibliography

ARE (2021). Landschaftstypologie Schweiz. https://www.are.admin.ch/are/de/home/laendliche-raeume-und-berggebiete/grundlagen-und-daten/landschaftstypologie-schweiz.html. Accessed: 2021-12-28.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561.

Bue, B. D. and Stepinski, T. F. (2006). Automated classification of landforms on mars. *Comput. Geosci.*, 32:604–614.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2019). Deep clustering for unsupervised learning of visual features.

Catani, F. (2020). Landslide detection by deep learning of non-nadiral and crowdsourced optical images. *Landslides*, 18.

Dong, S., Xu, H., Zhu, X., Guo, X., Liu, X., and Wang, X. (2020). Multi-View Deep Clustering based on AutoEncoder. In *Journal of Physics Conference Series*, volume 1684 of *Journal of Physics Conference Series*, page 012059.

Drăguţ, L. and Blaschke, T. (2006). Automated classification of landform elements using object-based image analysis. *Geomorphology*, 81(3):330–344.

Du, L., You, X., Li, K., Meng, L., Cheng, G., Xiong, L., and Wang, G. (2019). Multi-modal deep learning for landform recognition. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158:63–75.

EDA (2021). Die Schweiz entdecken - Geografie. https://www.eda.admin.ch/aboutswitzerland/de/home/umwelt/geografie.html. Accessed: 2021-12-28.

Ehsani, A. and Quiel, F. (2009). A semi-automatic method for analysis of landscape elements using shuttle radar topography mission and landsat etm+ data. *Computers Geosciences*, 35:373–389.

Gansbeke, W. V., Vandenhende, S., Georgoulis, S., Proesmans, M., and Gool, L. V. (2020). Scan: Learning to classify images without labels.

Guo, X., Liu, X., Zhu, E., and Yin, J. (2017). Deep clustering with convolutional autoencoders. In *Proceedings of International Conference on Neural Information Processing (ICONIP 2017)*, pages 373–382.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

Karimpouli, S. and Tahmasebi, P. (2019). Segmentation of digital rock images using deep convolutional autoencoder networks. *Computers Geosciences*, 126:142–150.

Li, F., Qiao, H., and Zhang, B. (2018). Discriminatively boosted image clustering with fully convolutional auto-encoders. *Pattern Recognition*, 83:161–173.

Li, F., Qiao, H., Zhang, B., and Xi, X. (2017). Discriminatively boosted image clustering with fully convolutional auto-encoders.

Li, S., Xiong, L., Tang, G., and Strobl, J. (2020). Deep learning-based approach for landform classification from integrated data sources of digital elevation model and imagery. *Geomorphology*, 354:107045.

Mai Ngoc, K. and Hwang, M. (2020). Finding the best k for the dimension of the latent space in autoencoders. In Nguyen, N. T., Hoang, B. H., Huynh, C. P., Hwang, D., Trawiński, B., and Vossen, G., editors, *Computational Collective Intelligence*, pages 453–464, Cham. Springer International Publishing.

Mohammadimanesh, F., Salehi, B., Mahdianpari, M., Gill, E., and Molinier, M. (2019). A new fully convolutional neural network for semantic segmentation of polarimetric sar imagery in complex land cover ecosystem. *ISPRS Journal of Photogrammetry and Remote Sensing*, 151.

Nousi, P. and Tefas, A. (2020). Self-supervised autoencoders for clustering and classification. *Evolving Systems*, 11.

Stepinski, T. F., Ghosh, S., and Vilalta, R. (2006). Automatic recognition of landforms on mars using terrain segmentation and classification. In Todorovski, L., Lavrač, N., and Jantke, K. P., editors, *Discovery Science*, pages 255–266, Berlin, Heidelberg. Springer Berlin Heidelberg.

Valentine, A. P., Kalnins, L. M., and Trampert, J. (2013). Discovery and analysis of topographic features using learning algorithms: A seamount case study. *Geophysical Research Letters*, 40(12):3048–3054.

Valentine, A. P. and Trampert, J. (2012). Data space reduction, quality assessment and searching ofseismograms: autoencoder networks for waveform dat. *Geophysical Journal International*, 189(2):1183–1202.

Yamada, T., Prügel-Bennett, A., and Thornton, B. (2021). Learning features from georeferenced seafloor imagery with location guided autoencoders. *Journal of Field Robotics*, 38(1):52–67.

Zhao, W., Xiong, L., Ding, H., and Tang, G. (2017). Automatic recognition of loess landforms using random forest method. *Journal of Mountain Science*, 14:885–897.

A Declaration of Originality

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Deep Feature Learning for Digital Elevation Models with Auto-encoders

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Spannagel	Reto

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 30.01.2022	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.