



Self-Supervised Learning for Semantic Segmentation of Archaeological Monuments in DTMs

RESEARCH ARTICLE

BASHIR KAZIMI

MONIKA SESTER

*Author affiliations can be found in the back matter of this article

ubiquity press

ABSTRACT

Deep learning models need a lot of labeled data to work well. In this study, we use a Self-Supervised Learning (SSL) method for semantic segmentation of archaeological monuments in Digital Terrain Models (DTMs). This method first uses unlabeled data to pretrain a model (pretext task), and then fine-tunes it with a small labeled dataset (downstream task). We use unlabeled DTMs and Relief Visualizations (RVs) to train an encoder-decoder and a Generative Adversarial Network (GAN) in the pretext task and an annotated DTM dataset to fine-tune a semantic segmentation model in the downstream task. Experiments indicate that this approach produces better results than training from scratch or using models pretrained on image data like ImageNet. The code and pretrained weights for the encoder-decoder and the GAN models are made available on Github.¹

CORRESPONDING AUTHOR:

Bashir Kazimi

Leibniz University Hannover
Institute of Cartography and
Geoinformatics Appelstr. 9a,
30167 Hannover, DE
b.kazimi@fz-juelich.de

KEYWORDS:

Self-Supervised Learning;
Digital Terrain Models; Deep
Learning; Archaeology;
Convolutional Neural
Networks; Generative
Adversarial Networks; Relief
Visualization

TO CITE THIS ARTICLE:

Kazimi, B and Sester, M. 2023.
Self-Supervised Learning for
Semantic Segmentation of
Archaeological Monuments
in DTMs. *Journal of Computer
Applications in Archaeology*,
6(1): 155–173. DOI: [https://doi.
org/10.5334/jcaa.110](https://doi.org/10.5334/jcaa.110)

1. INTRODUCTION

In recent years, the field of computational archaeology has witnessed remarkable advancements, with Deep Learning (DL) playing a pivotal role in reshaping our understanding of ancient civilizations. DL is a subfield of machine learning characterized by the utilization of neural networks to discern intricate patterns and representations from data, often surpassing human performance in tasks such as image classification (Voulodimos et al., 2018), object detection (Zhao et al., 2019), natural language processing (Stahlberg, 2020), and medical image analysis (Kumar and Bindu, 2019). However, it requires a lot of labeled data. To overcome this limitation, a common approach is to use pretrained models on larger datasets in the same domain and fine-tune them for the task with limited annotations. Trained models refer to neural networks that have undergone a learning process, adapting to specific datasets and tasks. In contrast, pretrained models are neural networks initially trained on extensive datasets, providing a foundation for further fine-tuning on specialized tasks.

Many image datasets with a lot of labeled examples, such as ImageNet (Deng et al., 2009), COCO-Stuff (Caesar et al., 2018), ADE20K (Zhou et al., 2017), and Pascal VOC (Everingham et al., 2015), are available. Researchers use pretrained deep learning models on these datasets to fine-tune them for tasks such as classification, object detection and semantic segmentation. This approach results in better performance and faster convergence when annotations are limited, than training a model from scratch.

Self-Supervised Learning (SSL) is a useful technique when labeled data is scarce. It involves two steps: in the first step, called the pretext, a model is pretrained using unlabeled data and an inherent characteristic or derivative of the data as an implicit supervision signal. In the second step, called the downstream, the model is fine-tuned using labeled data, initialized with the pretrained weights from the pretext step, to solve a supervised problem. This approach makes the model learn hidden representations and useful features in the data that can be transferred and used in downstream supervised tasks.

There are many ways to pose the pretext task in SSL as a supervised problem using implicit supervision signals for unlabeled data. Examples include training a model to predict the rotation of images (Gidaris et al., 2018), or to predict the relative positioning of two patches from a 3x3 image grid (Doersch et al., 2015). Another example is using an autoencoder to encode and reconstruct a given input (Kazimi et al., 2020a). Examples of supervised downstream tasks include image classification, object detection, and semantic segmentation. These tasks can be trained from scratch or fine-tuned, using weights from

models pretrained on annotated data or those pretrained in the context of SSL pretext on unlabeled data.

Researchers in the image domain often use models pretrained on large annotated datasets, such as ImageNet, COCO-Stuff, ADE20K, and Pascal VOC, to fine-tune supervised models for downstream tasks when large annotated datasets for a specific task are not available (Krishna and Kalluri, 2019). Similarly, due to the lack of benchmarked annotated Digital Terrain Model (DTM) datasets, researchers working with DTM data also use models pretrained on image data (Bundzel et al., 2020; Øivind Due Trier et al., 2021). In this research, one goal is to evaluate the role of pretrained data types: thus besides image data, also DTM data are used. To this end, different so-called Relief Visualizations (RVs) (Kokalj and Hesse, 2017) are used as implicit supervision signal. Thus, unlabeled DTM data and the RVs are used to pretrain an encoder-decoder model and a Generative Adversarial Network (GAN) in the pretext step. Pretrained weights are then used to fine-tune a semantic segmentation model on a small annotated DTM dataset. This leads to better results compared to training from scratch or using weights from models pretrained on image data.

In the downstream step, a supervised model is initialized with the pretrained weights and fine-tuned on annotated DTM data for semantic segmentation of archaeological monuments in the Harz region in Lower Saxony. The overall structure of this research is shown in Figure 1 and the contributions are summarized as follows:

- Exploring and highlighting the potentials of deep learning in detecting archaeological structures which are difficult to identify (e.g., eroded and/or partially damaged burial mounds) or complex to describe (e.g., mining holes).
- Sharing encoder-decoder and GAN models pretrained on DTM data which can be transferred to supervised downstream tasks such as classification, object detection, and semantic and instance segmentation.
- Highlighting that compared to training deep learning models from scratch or initializing them with random weights or pretrained weights from models trained on natural images, using pretrained weights from encoder-decoder and GAN models pretrained on DTM data provides advantages and is a promising approach to improve the performance of deep learning models on tasks with DTM data.

2. RELATED WORK

With the increasing success in application of deep learning techniques in many fields, researchers in archaeology are also using deep learning in their tasks. Kazimi et al. (2018) and Politz et al. (2018) trained a CNN

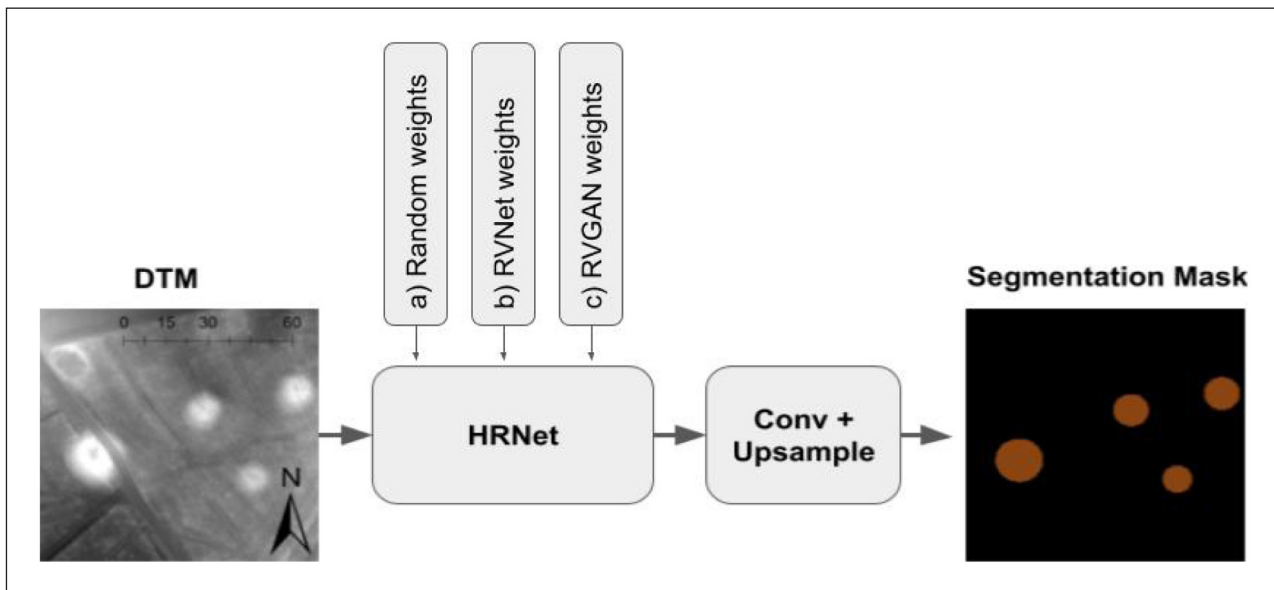


Figure 1 Overall structure of this research. The HRNet model is trained for semantic segmentation of archaeological structures and three different methods (a, b, c) are used to initialize the model parameters. RVNet and RVGAN are encoder-decoder and Generative Adversarial Network based models pretrained on unlabeled DTM data.

classifier to detect tracks, streams, and lakes using DTM inputs. Based on the proposal by Du et al. (2019) that a combination of different geomorphological information can help improve the performance of deep learning models, Kazimi et al. (2020b) trained a modified High Resolution Network (HRNet) that takes multiple inputs, including DTM, SLRM, LD, SVF, openness, and slope, to detect archaeological terrain structures. Soroush et al. (2020) and Bundzel et al. (2020) trained CNN models on satellite imagery and DTM data to detect qanat shafts and ancient Maya buildings. Other applications of deep learning in archaeology include tomb and burial mound classification (Caspari and Crespo, 2019; Guyot et al., 2018), archaeological monument segmentation (Kazimi et al., 2019), and extraction of terrain structures (Satari et al., 2021).

Deep learning is commonly adopted in many research fields, but one major problem it faces is the lack of annotated datasets. To overcome this issue, researchers use models pretrained on image data and fine-tune them on their own datasets with limited annotations. This technique has proven to be effective despite the domain gap between the image data and DTMs. Examples include using the AlexNet model (Krizhevsky et al., 2012) for automated mapping of charcoal kilns (Trier et al., 2018), using a Faster R-CNN model (Ren et al., 2015) for detection of archaeological objects in the Netherlands (Verschoof-van der Vaart and Lambers, 2019) and mapping cultural heritage in Norway (Trier et al., 2021), and using a modified version of Mask R-CNN (He et al., 2017) for detection of archaeological sites in the North German Lowland (Bonhage et al., 2021). Other examples of fine-tuning models for tasks in archaeological research include classifying ancient Maya structures (Somrak et al., 2020), mapping archaeological topography on Arran,

Scotland (Trier et al., 2019), and detecting valley fills in DTMs (Maxwell et al., 2020).

To alleviate the domain-gap problem and use models pretrained on the same data, i.e., DTMs, SSL can be utilized. As explained in Section 1, SSL consists of two steps: pretext and downstream. In the first step, a model is trained on unlabeled data, and in the second step, the pretrained weights are used to fine-tune the model on labeled data for a specific task. As a pretext task, Noroozi and Favaro (2016) trained a model to solve jigsaw puzzles for images. They randomly crop a 225×225 pixel window and divide it into a 3×3 grid. A random 64×64 pixel tile from each of the 9 grid cells is selected and randomly reordered. The randomly reordered tiles are fed to a model which is trained to learn the permutation order of the 9 tiles. Such pretraining leads to better performance when transferred to supervised tasks such as classification, detection and segmentation on the Pascal VOC dataset. Other examples of pretext tasks include image colorization (Zhang et al., 2016), stacked autoencoder for DTMs (Kazimi et al., 2020a), and image-rotation identification (Doersch et al., 2015).

Two methods, encoder-decoder and GANs, are used and compared in the pretext tasks in this research. Encoder-decoder models transform an input into an embedding matrix/vector which is then used to reconstruct the original input or another representation of it. GANs, first introduced by Goodfellow et al. (2014), are generative approaches that frame a task with unlabeled data as a supervised learning problem. A GAN architecture consists of a generator and a discriminator model. The generator samples random noise and generates plausible examples for the task domain, while the discriminator separates real examples from those generated by the generator. Examples of encoder-

decoder models include Guo et al. (2017), Masci et al. (2011), and David and Netanyahu (2016). Examples of GANs include image-to-image translation (Isola et al., 2017; Zhu et al., 2017), text-to-image translation (Zhang et al., 2017; Zhu et al., 2017), video generation (Vondrick et al., 2016), photo blending (Wu et al., 2019), inpainting (Pathak et al., 2016), image super-resolution (Ledig et al., 2017), and cartoon generation (Jin et al., 2017).

3. METHODOLOGY

The goal of this research is two-fold: utilizing deep learning for semantic segmentation of archaeological monuments and creating pretrained deep learning models using unlabeled DTMs so that they are transferable to any supervised downstream task with annotated DTM datasets. Therefore, the task is framed as a Self-Supervised Learning (SSL) problem which consists of two steps: pretext and downstream, explained as follows.

3.1 PRETEXT

Pretext is the first step in SSL which utilizes unlabeled data but exploits labeling that can easily and automatically be obtained from the structure of the data (Doersch et al., 2015; Noroozi & Favaro, 2016). To reveal the inherent characteristic of the data, RVs are used. These derivatives of the DTMs are typically used to enhance special structures in the terrain mainly for visual inspection and analysis. In this research, the following visualizations are used:

- **Simple Local Relief Model (SLRM)** is a technique used to highlight small-scale features in DTM data. It involves creating a trend removal map by smoothing the DTM with a low-pass convolution filter and subtracting it from the original DTM, then creating a purged DTM by creating zero contours in the trend removal map and interpolating the points. The final raster, called the SLRM, is created by subtracting the purged DTM from the original one, and contains a less distorted representation of small-scale features.
- **Local Dominance (LD)** indicates how dominant an observer would be from a given point compared to its neighboring points (Hesse, 2016). The dominance value for each point is calculated using the average angle at which a virtual observer standing at that

point would look down at the neighboring points within a fixed radius r . This gives pixels at local peaks high dominance values and make them appear brighter, while pixels at local sinks have small dominance values and appear darker. LD is suitable for visualizing protruding features such as burial mounds and sunken features such as hollow ways.

- The **Sky View Factor (SVF)** value for a point is calculated relative to surrounding points within a given radius to show what portion of the sky is visible. SVF is well suited for archaeological structures, such as mining holes (Kokalj and Hesse, 2017).
- **Slope** is related to the first derivative and indicates the steepness of a surface. It is calculated as the maximum rate of change of the elevation of a point with respect to its neighboring points (Gelbman and Papo, 1984).
- **Hillshade** is a way to show terrain surface based on shadows from a light source, usually from the northwest. Pixels perpendicular to light source get high value and those at angle greater than 90 get low value (Kokalj and Hesse, 2017). RGB hillshade is created using hypothetical light sources from three directions.

In this research, two different architectures are used in the pretext. The first one is an encoder-decoder approach that takes an input DTM and learns to predict the corresponding RVs such as LD, SLRM, slope and SVF. Hence, the encoder-decoder architecture is hereafter referred to as the Relief Visualization Network (RVNet). The second approach is based on Generative Adversarial Networks (GANs) in which the generator is trained to take DTM inputs and generate realistic RVs to fool the discriminator, and the discriminator is trained to distinguish between the generated RVs and the original RVs calculated using the Relief Visualization Toolbox (RVT) (Zakšek et al., 2011; Kokalj and Somrak, 2019; Kokalj and Hesse, 2017). Hence, the GAN-based architecture is hereafter referred to as the Relief Visualization GAN (RVGAN). Both approaches are explained in details as follows.

3.1.1 Relief Visualization Network (RVNet)

RVNet is an encoder-decoder architecture that takes input DTMs and predicts corresponding RVs like LD, SLRM, slope and SVF. The encoder part of the model is based on HRNet (Figure 2), and the decoder has convolution and

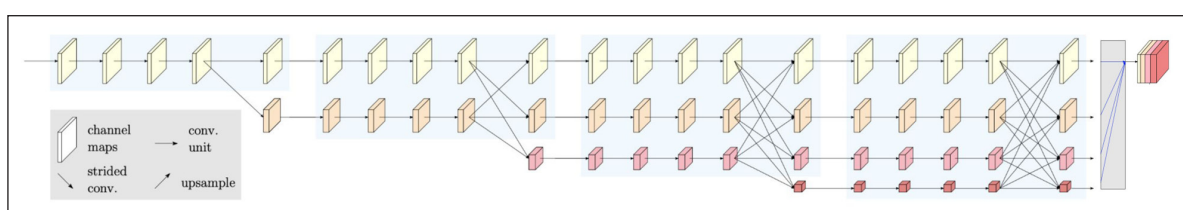


Figure 2 HRNet model (Sun et al., 2019): the backbone for all the methods in this research.

upsampling layers to match output dimensions with the original RVs and adjust the number of expected outputs. RVNet is defined by Equation 1.

$$\hat{y} = \text{Decode}(\text{HRNet}(x)) \tag{1}$$

Where x denotes a DTM patch, \hat{y} denotes the predicted RVs, and **HRNet** and **Decode** are the encoder and decoder part of the RVNet model illustrated in Figure 3. It is trained using the Mean Squared Error (MSE) function which is also referred to as the quadratic or L_2 loss in literature. The MSE between the predicted RVs \hat{y} and the target/expected RVs y can be calculated using Equation 2.

$$\text{MSE} = \frac{1}{N \times M \times K} \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^K (\hat{y}_{ijk} - y_{ijk})^2 \tag{2}$$

Where N and M denote the spatial dimensions of the RVs, K denotes the number of output channels, i.e., number of different RVs predicted for a given input, and \hat{y} and y represent the predicted and target RVs, respectively.

Once trained, the **HRNet** part of the RVNet architecture can be used as a fixed feature extractor or fine-tuned for supervised downstream tasks such as classification, semantic segmentation and instance segmentation with annotated data. In this paper, it is used for semantic segmentation. This technique, i.e., fixed-feature extraction or fine-tuning, leads to a better performance than training from scratch, as discussed in Section 3.2.

3.1.2. Relief Visualization GAN (RVGAN)

RVGAN is based on the GAN architecture (Goodfellow et al., 2014), specifically the conditional Pix2Pix GAN by Isola et al. (2017). The goal is to train a generator that predicts realistic Relief Visualizations (RVs) for a given DTM input and fools the discriminator into thinking the

RVs are real. Similar to the RVNet architecture explained previously, the **HRNet** architecture combined with the convolutional and upsampling layers (**Decoder**) are used as the generator part of the RVGAN. The discriminator is a model made of 3 convolutional layers with leaky ReLU activations after each layer (except the last layer). The discriminator takes a DTM along with either the real or the generated RVs and is trained to detect whether the given RVs are fake or real. The discriminator used here is the so-called PatchGAN discriminator. It is a type of discriminator that penalizes the structure at the scale of local image patches, i.e., rather than trying to classify the whole input image, it classifies each $N \times N$ patch in an image into real or fake (Isola et al., 2017). N is set to 70 in this research, meaning the discriminator is a PatchGAN discriminator of size 70×70 . Each pixel in its output indicates whether the corresponding 70×70 pixel patch in the input is real or fake. The RVGAN model is illustrated in Figure 4 and is trained using the loss function \mathcal{L} defined in Equation 3.

$$\mathcal{L}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))] + \lambda \mathcal{L}_p(G) \tag{3}$$

Where $\log D(x,y)$ denotes the predicted probability by the discriminator, D , that the RVs, y , are real. $G(x,z)$ represents the generated fake RVs by the generator, G , given the input DTM, x . $\mathbb{E}_{x,y}$ and $\mathbb{E}_{x,z}$ are the expected values over all examples. L_p denotes the p^{th} norm between the generated and expected RVs.

The **HRNet** part of the RVGAN architecture can also be used a fixed feature extractor or fine-tuned for supervised downstream tasks with annotated datasets. The improvement in performance using the pretrained RVGAN is even more significant compared to the RVNet model, as discussed in Section 3.2.

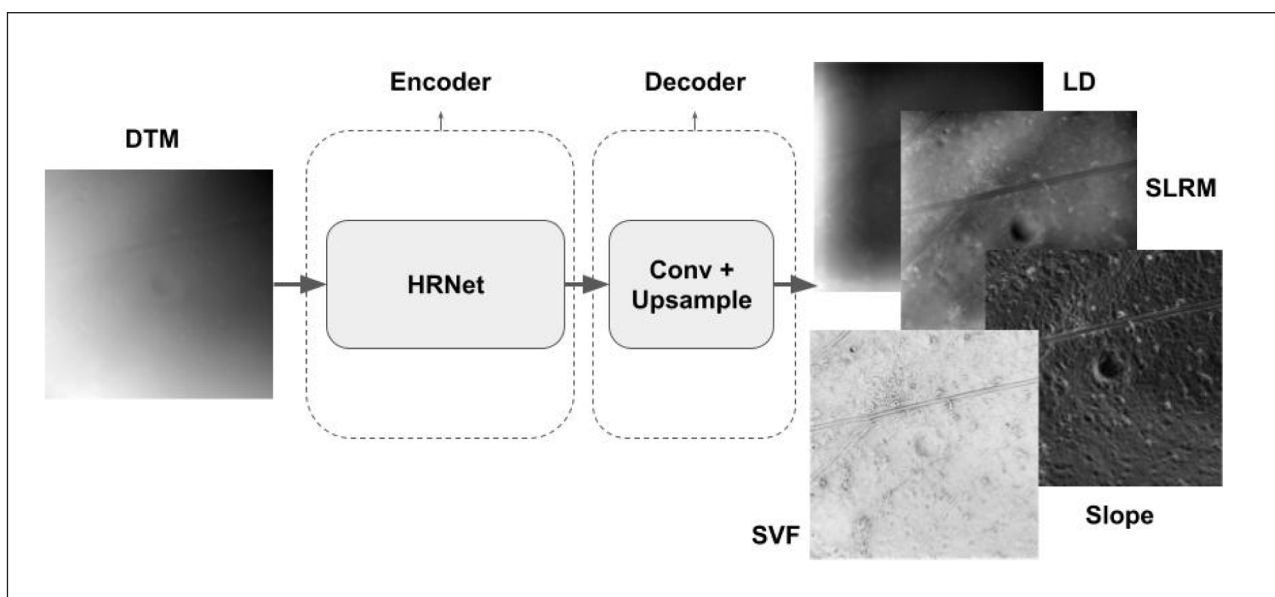


Figure 3 Architecture of the RVNet. The encoder part is the HRNet model shown in Figure 2. The decoder consists of convolutional layers to adjust the number of outputs and upsampling layer to match the spatial dimension of the outputs.

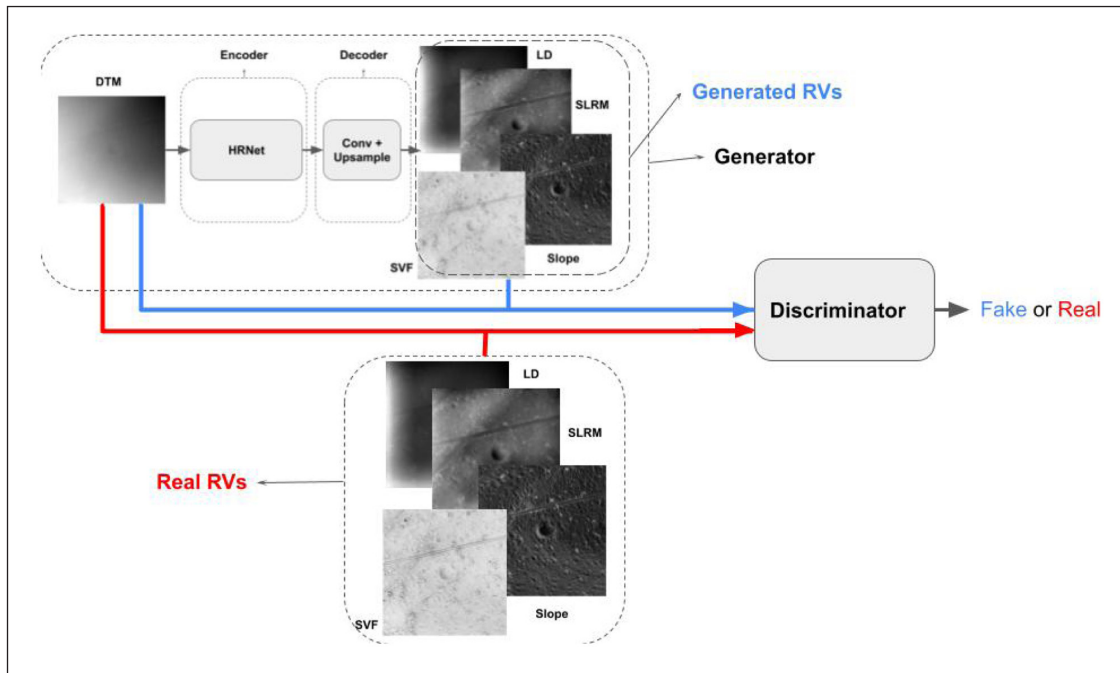


Figure 4 Architecture of the RVGAN. The generator is the same architecture as the RVNet shown in Figure 3. The discriminator is made of 3 convolutional layers with leaky ReLU activations after each layer (except the last layer).

3.2 DOWNSTREAM

The supervised downstream task in this research is semantic segmentation. To study the impact of pretraining in the pretext step, the model of choice for backbone here is also the HRNet architecture. The model is similar to the RVNet, but the decoder part is altered and adapted for semantic segmentation, i.e., the number of outputs for the last convolutional layer is set to the number of categories in the annotated dataset. The architecture is illustrated in Figure 1. It is trained on a small annotated Digital Terrain Model (DTM) dataset of archaeological monuments. The loss function for training is the Cross Entropy (CE) function shown in Equation 4.

$$CE = - \sum_{i \in I} \sum_{c \in C} y_{i,c} \log \hat{y}_{i,c} \quad (4)$$

Where I denotes the image pixels, $y_{i,c}$ indicates whether the i^{th} pixel is in category c , and $\hat{y}_{i,c}$ shows the predicted probability that pixel i belongs to category c .

The semantic segmentation model uses the **HRNet** module, shown in Figure 1, with random weight initialization, or fine-tuned with pretrained weights from RVNet and RVGAN. To study the impact of pretraining with DTM data compared to data from other domains, e.g., natural images, the model is also fine-tuned after initializing the **HRNet** module with pretrained ImageNet, COCO-Stuff, Pascal VOC, and ADE20K weights.

4. EXPERIMENTS AND RESULTS

Experiments were conducted on unlabeled DTM data in the pretext step using RVNet and RVGAN. Details of the

dataset and training procedure for each model are given in Section 4.1. For the supervised downstream task, i.e., semantic segmentation, a small annotated DTM dataset of archaeological monuments was used. The dataset and training details are given in Section 4.2.

4.1 PRETEXT EXPERIMENTS

4.1.1 Experiments with RVNet

RVNet was trained to predict RVs, such as LD, SLRM, slope, and SVF for a given DTM input. The DTM data was created from Airborne Laser Scanning (ALS) or LiDAR data from Lower Saxony which has a resolution of 0.5 meters per pixel and covers 47,000 km². A hillshade RV of the data is shown in Figure 5. To create the dataset for pretext experiments, 200,000 DTM patches of 224 × 224 pixels were randomly cropped from the region. The RVT Toolbox (Zakšek et al., 2011; Kokalj and Somrak, 2019; Kokalj and Hesse, 2017) was used to calculate RVs for each patch and the RVNet was trained to predict RVs that are as similar as possible to the calculated RVs using the RVT toolbox. Example DTMs and RVs are shown in Table 1.

The 224 × 224 pixel DTMs and the corresponding RVs were normalized in the range of 0 to 1 using Equation 5 as follows.

$$X = \frac{X - \text{MIN}(X)}{\text{MAX}(X) - \text{MIN}(X)} \quad (5)$$

Where X is a 224 × 224 pixel DTM patch or an RV, and MIN and MAX are the minimum and maximum operations, respectively.

The dataset was divided into 180,000 training, 10,000 validation and 10,000 testing examples. The model was trained using Python and the PyTorch deep learning

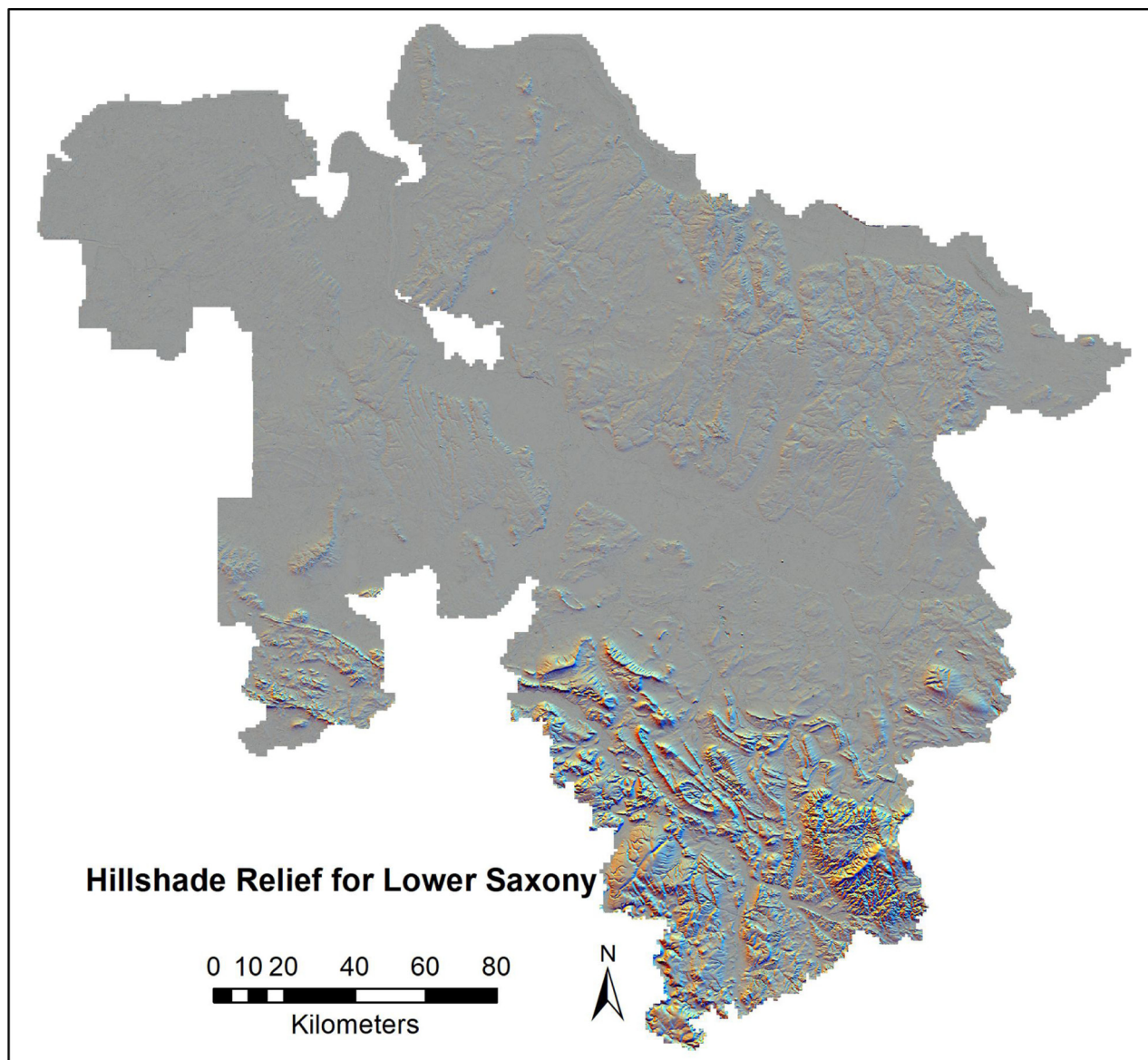


Figure 5 Hillshade RV for DTM data from Lower Saxony.

library (Paszke et al., 2019) for 200 epochs. MSE was used as the objective function optimized by the Stochastic Gradient Descent (SGD) function with a starting learning rate of 0.001, a momentum of 0.9, and a weight decay of 0.0001. The batch size was set to 208 examples during training and data augmentations such as vertical and horizontal flipping, random rotation by 90 degrees, and cropping 128×128 windows from the 224×224 pixel patches were applied. The training history showing the MSE scores for training and validation data is plotted in Figure 6. The model weights scoring the best, i.e., minimum MSE score on validation data were saved (epoch 139 in this case).

4.1.2 Experiments with RVGAN

RVGAN, similar to RVNet, was trained on Digital Terrain Models (DTMs) and their corresponding RVs, which were normalized to 0-1 range. The model was trained using the objective function in Equation 3 ($\lambda = 100$, $p = 1$, meaning L_1 norm was used for the generator) and optimized using

Adam (Kingma and Ba, 2015) optimizer ($\text{lr} = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.999$). Data augmentation techniques like flipping, rotation, and cropping were applied with a batch size of 256. MSE was used as the objective function for the discriminator. The model was trained for 100 epochs and the generator weights scoring best on validation data were saved.

The best RVNet model (epoch 139) and the best generator in RVGAN (epoch 51) were used to predict the RVs for the test data and the corresponding MSE and L_1 errors are listed in Table 2. An example prediction by both models is shown in Table 3. While RVNet was only trained to minimize the difference between the real RVs and its own predictions, RVGAN was additionally tasked with generating realistic predictions to fool the discriminator. This translated into better L_1 and L_2 scores and also better predictions by the RVGAN compared to the RVNet model.

4.2 DOWNSTREAM EXPERIMENTS

For the downstream task, a small annotated DTM dataset of archaeological monuments in the Harz region in Lower

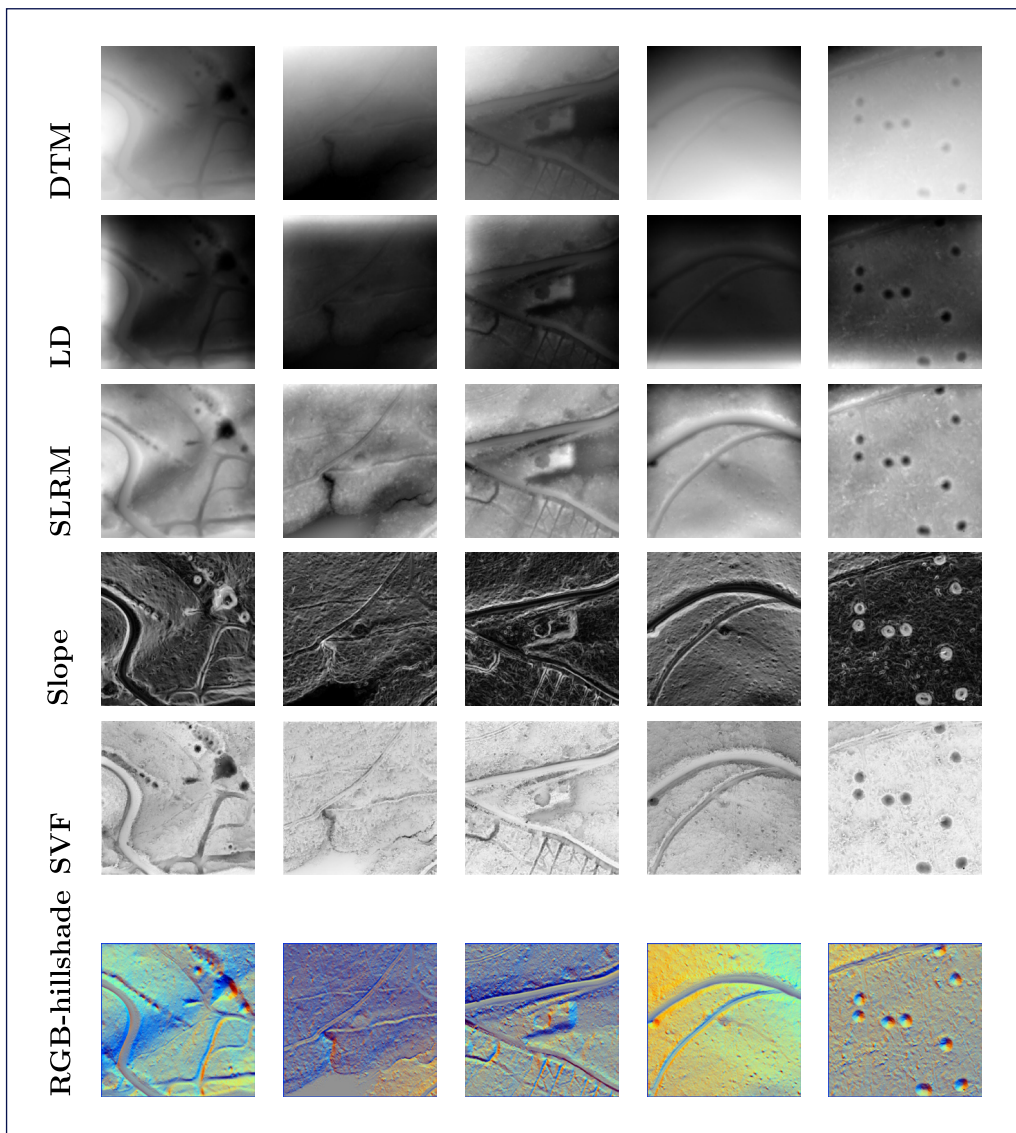


Table 1 Example DTMs and corresponding RVs.

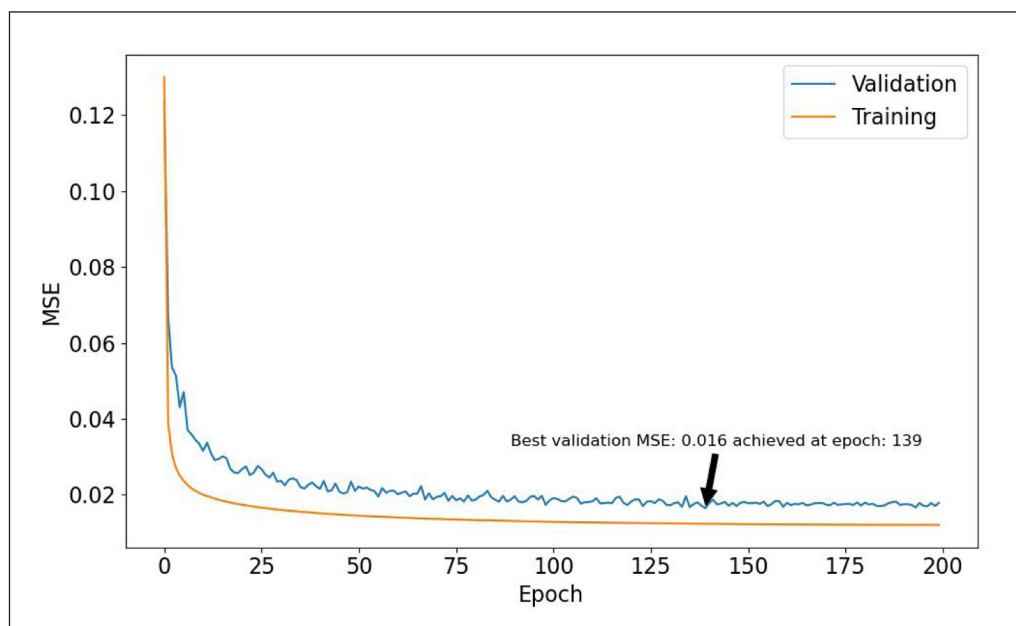


Figure 6 Training history for the RVNet.

Saxony was used. The dataset contains annotated examples of 4 kinds of structures including Bomb Craters (BC), Charcoal Kilns (CK), Burial Mounds (BM) and Mining Holes (MH). Information about the annotated structures are listed in Table 4 and examples of annotations created using the ArcGIS software are shown in Figure 7.

The annotated dataset was split into training, validation and test sets as shown in Table 5. For each monument, a 256×256 DTM window was cropped, and a corresponding segmentation mask was created using ArcGIS. An example DTM and segmentation mask containing burial mounds is shown in Figure 8. For each DTM, RVs such as RGB hillshade, LD, SLRM, slope and SVF were calculated using the RVT Toolbox. The model was trained using the DTMs and RVs separately and combined, and the results were compared. Different optimization functions were used, including Adam, SGD, and RMSProp, and the training was run for 100 epochs with batch size of 96. Data augmentations such as flipping, rotation, and cropping 128×128 windows from the 256×256 pixel patches were also applied. The objective function was the CE function in Equation 4 and the evaluation metric was the Intersection over Union (IoU) score, also known as the Jaccard Index, shown in Equation 6.

$$\text{Jaccard}(y, \hat{y}) = \text{IoU}(y, \hat{y}) = \frac{\|y \cap \hat{y}\|}{\|y \cup \hat{y}\|} \tag{6}$$

Where y and \hat{y} denote the ground truth and predicted output, respectively. For each choice of input data and experimental setup, the best results on the test data are reported in Table 6.

As observed in Table 6, the combination of four RVs, i.e., {LD, Slope, SLRM, SVF} leads to the best mIoU score. Therefore, these four RVs were used as the supervision signals for the Relief Visualization Network (RVNet) and Relief Visualization GAN (RVGAN) in the pretext step as well. The idea is that since these four RVs are the most informative among other individual RVs or their combinations, deep learning models can be pretrained to learn computing them given an input DTM. A model that can learn to compute them is thought to have learned the structure and hidden characteristics of the dataset well enough to be used for fine-tuning on supervised tasks. To prove this, the model in Figure 1 was initialized with the pretrained weights of the RVNet and RVGAN and fine-tuned for semantic segmentation using DTMs as the input. In fine-tuning, a common practice is to freeze the weights in some layers of the model. In the HRNet model, there are 4 stages as shown in Figure 1, and experiments were conducted freezing layers from the first layer up to the each of these stages. The rest of the experimental setup was kept the same as the the previous semantic segmentation experiments with different RVs. The configurations with the best results are listed in Table 7.

In order to compare and study the effect of pretraining with a dataset in the same domain, i.e., DTMs, and a

	RVNET	RVGAN
L_1	0.0929	0.0390
L_2	0.0170	0.0042

Table 2 L_1 and L_2 losses on test data by RVNet and RVGAN.

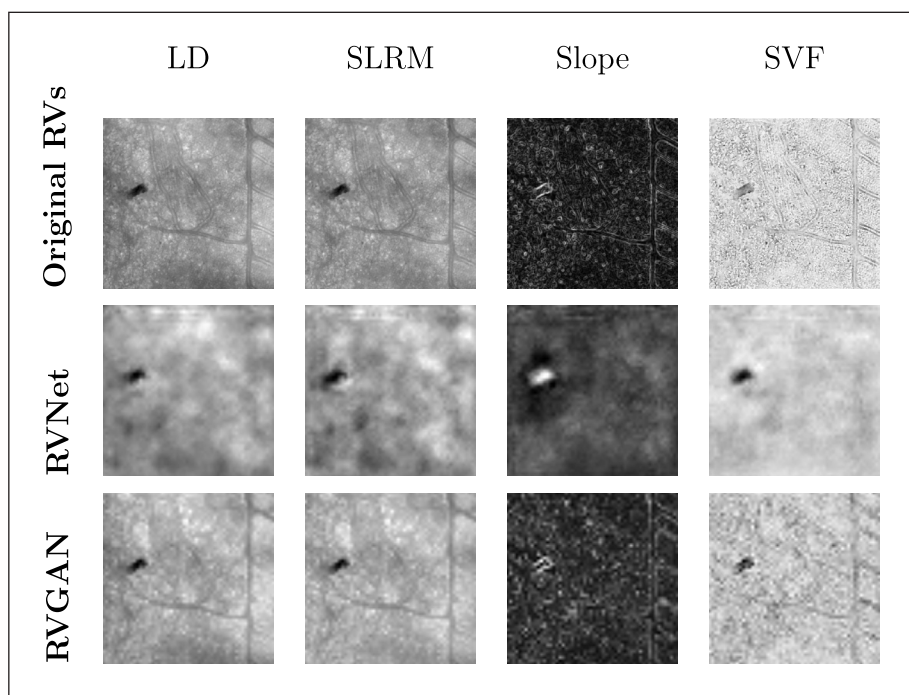


Table 3 Example prediction by RVNet and RVGAN. As observed, RVGAN made better and less blurry predictions compared to RVNet. It is intuitive as the generator in RVGAN was trained to not only generate the RVs, but also fool the discriminator.

MONUMENT	NO. EXAMPLES	MIN. Ø	AVG. Ø	MAX. Ø
Bomb Craters	617	1.3 m	7.4 m	38 m
Charcoal Kilns	2543	6.3 m	15.3 m	24.4 m
Burial Mounds	1410	4.5 m	14.8 m	37.7 m
Mining Holes	2986	1.2 m	8 m	63 m

Table 4 Statistics for the annotated dataset. Ø denotes the diameter.

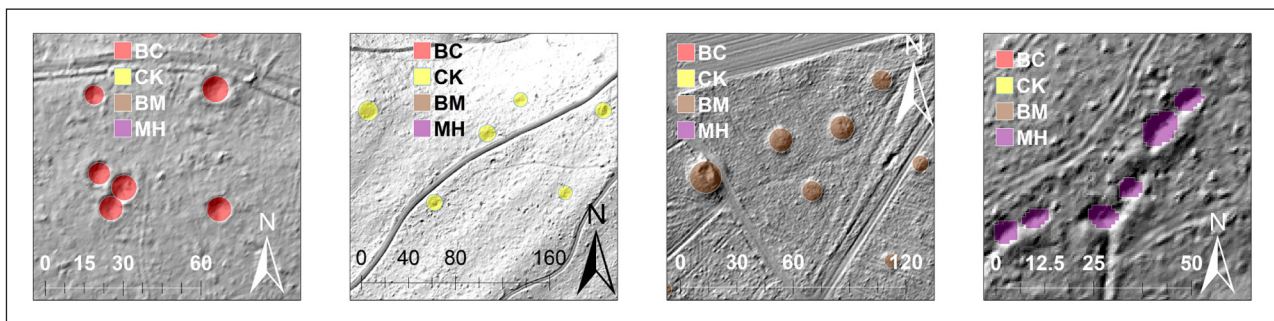


Figure 7 Example annotations for the dataset.

SPLIT	BOMB CRATERS	CHARCOAL KILNS	BURIAL MOUNDS	MINING HOLES
Training	314	1560	833	1741
Validation	169	479	357	481
Testing	134	504	220	764

Table 5 Three different, non-overlapping regions are selected for training, validation and test set. 3351 examples where no monuments exist and include only background pixels were also included in the training set.

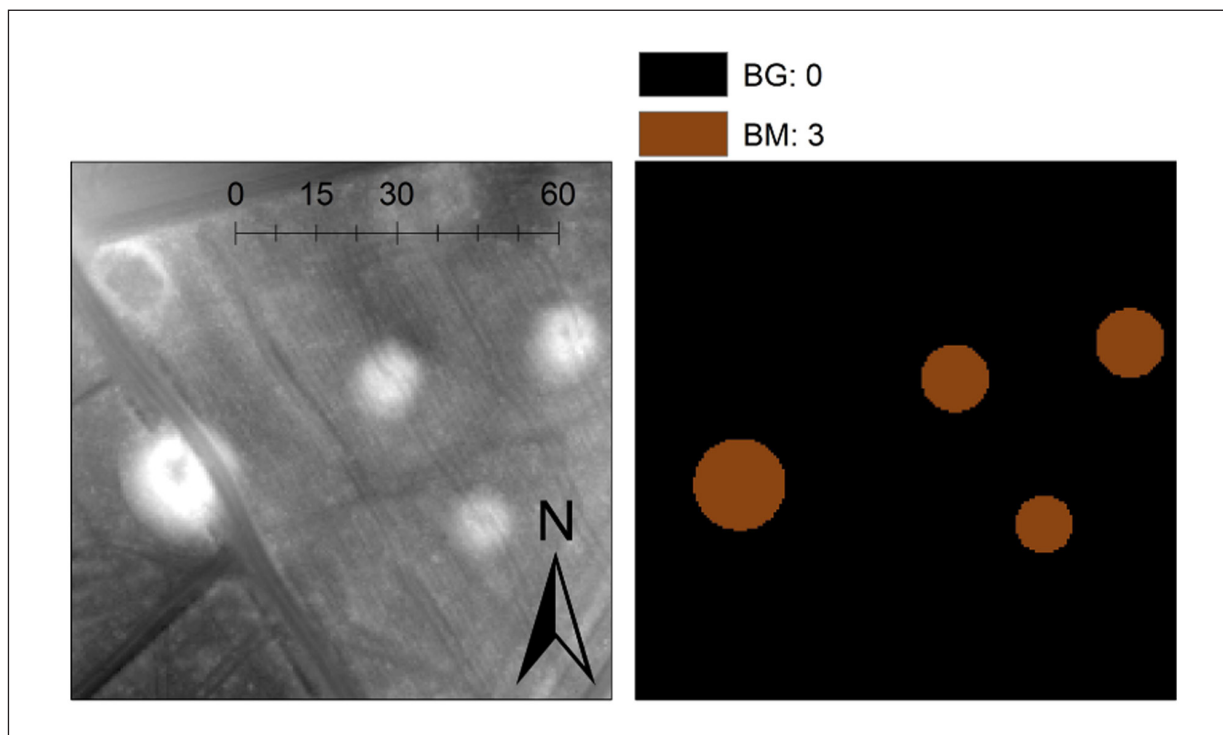


Figure 8 Example DTM input and the corresponding mask showing Background (BG) labeled as 0 and Burial Mounds (BM) labeled as 3 for semantic segmentation.

INPUT	OPTIMIZER	MIOU	BC IOU	CK IOU	BM IOU	MH IOU
Four	RMSProp	63.71	65.18	52.29	57.13	45.49
Five	RMSProp	63.44	64.54	57.25	53.05	43.86
DTM	RMSProp	62.64	60.15	54.37	53.65	46.56
SLRM	RMSProp	62.18	65.32	50.59	52.96	43.66
LD	Adam	61.91	59.98	58.37	50.40	42.29
All	RMSProp	61.29	62.55	49.13	54.79	41.63
SVF	Adam	61.23	63.44	52.23	46.08	46.00
RGB	SGD	61.03	55.14	55.57	60.64	35.31
Slope	SGD	59.27	59.50	50.52	52.16	35.95

Table 6 Training with random weight initialization. Four, Five and All refer to using combinations of {LD, Slope, SLRM, SVF}, {DTM, LD, Slope, SLRM, SVF}, and {DTM, LD, RGB, Slope, SLRM, and SVF}, respectively, as the model inputs. The best optimizer for each input choice is listed in the Optimizer column. The top mIoU score and individual IoU scores for Bomb Craters (BC), Charcoal Kilns (CK), Burial Mounds (BM) and Mining Holes (MH) are shown in **bold**.

WEIGHTS	OPTIMIZER	FROZEN	MIOU	BC IOU	CK IOU	BM IOU	MH IOU
Random	RMSProp	None	62.64	60.15	54.37	53.65	46.56
RVNet	RMSProp	None	63.02	61.52	58.44	50.42	46.25
RVGAN	RMSProp	2	63.18	61.96	56.35	50.24	48.87

Table 7 Training the semantic segmentation model using DTM inputs and random, RVNet, and RVGAN weight initialization. The top mIoU score and individual IoU scores for Bomb Craters (BC), Charcoal Kilns (CK), Burial Mounds (BM) and Mining Holes (MH) are shown in **bold**.

different domain, i.e., natural images, experiments were conducted by fine-tuning the semantic segmentation model using pretrained weights from the natural images domain, e.g., ImageNet, COCO-Stuff, ADE20K, and Pascal VOC datasets. Since the models pretrained on the previously mentioned natural images datasets expect input images with three channels, only the RGB hillshade RV was used in these experiments and the rest of the experimental setup was kept the same. Best results and their comparison to pretraining with DTM inputs are listed in Table 8.

As observed in Table 7, using pretrained weights of RVNet and RVGAN in fine-tuning improves mIoU scores compared to random weight initialization. RVGAN has a better impact on performance than RVNet, as RVGAN was trained to produce realistic Relief Visualizations (RVs) and fool the discriminator, providing an extra incentive to generate better RVs.

Table 8 compares the impact of pretraining with datasets of the same domain, i.e., DTMs and those of different domain, i.e., natural images. While using the pretrained weights from natural image datasets such as ImageNet, COCO-Stuff, ADE20K and Pascal VOC leads to a better score compared to random weight initialization with RGB hillshade as the inputs, the results are in general similar to training with random weight initialization and using DTM data as the input.

Using pretrained weights from the same domain, however, leads to the best scores as observed in the final two rows of the table which list the results of fine-tuning with RVNet and RVGAN weights. Even though the IoU scores for Bomb Craters (BC), Charcoal Kilns (CK), and Mining Holes (MH) are higher for models pretrained with ADE20K, COCO-Stuff and random weight initialization using RGB as the input, the scores are not equally good for other categories. In the case of pretraining with RVNet and RVGAN, the scores are overall stable for all categories as reflected in the mIoU scores.

To evaluate the results qualitatively, four different regions each containing examples of Bomb Craters (BC), Charcoal Kilns (CK), Burial Mounds (BM), and Mining Holes (MH) were selected as test regions. A sliding window approach was used to make predictions on each region by each model. The regions were scanned by cropping a window of 128×128 pixels starting from the top left and going right/down with a stride of 85 pixels (i.e., 85 pixels of overlap between successive windows) and making predictions. The final predictions for the test regions are shown in Figures 9–12. Examples are shown for the models trained after being initialized with random weights, RVNet weights, RVGAN weights, and ImageNet (the best performing model among pretrained weights from natural images).

INPUT	WEIGHTS	OPTIMIZER	MIOU	BC IOU	CK IOU	BM IOU	MH IOU
RGB	Random	SGD	61.03	55.14	55.57	60.64	35.31
RGB	ImageNet	RMSProp	62.85	61.33	58.84	52.56	43.23
RGB	COCO-Stuff	Adam	62.79	64.10	59.19	50.96	41.19
RGB	ADE20K	Adam	62.64	66.47	48.40	56.05	43.90
RGB	Pascal VOC	Adam	62.63	66.46	52.60	56.48	39.16
DTM	Random	RMSProp	62.64	60.15	54.37	53.65	46.56
DTM	RVNet	RMSProp	63.02	61.52	58.44	50.42	46.25
DTM	RVGAN	RMSProp	63.18	61.96	56.35	50.24	48.87

Table 8 Comparing the effects of pretrained weights from different domains. The top mIoU score and individual IoU scores for Bomb Craters (BC), Charcoal Kilns (CK), Burial Mounds (BM) and Mining Holes (MH) are shown in **bold**.

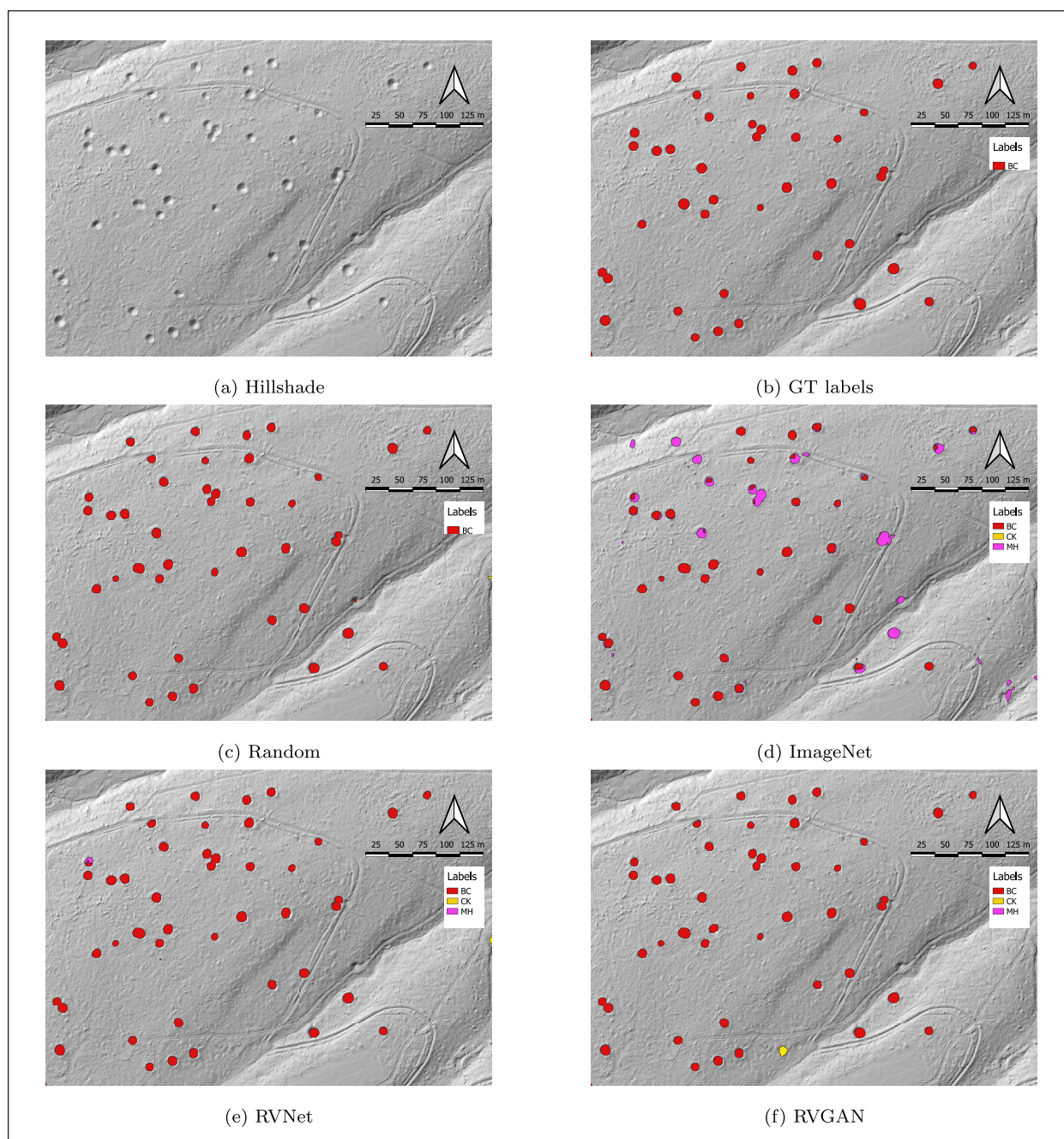


Figure 9 Example predictions for regions with Bomb Craters (BC).

As illustrated in Figure 9, random, pretrained RVNet, and pretrained RVGAN weights lead to similar performance in detecting bomb craters. ImageNet weights lead to the worse predictions among all of them, i.e., many bomb craters are falsely labeled as mining holes.

Figure 10 shows predictions for a region with charcoal kilns. The model initialized with ImageNet weights falsely labels some charcoal kilns examples as mining holes and makes a lot of false positive predictions. Similarly the model initialized with random weights predicts a lot of false positives even though it recovers most of the charcoal kiln examples. While the model initialized with pretrained RVGAN weights does not make false positive predictions, it still misses a few examples of charcoal kilns. The model initialized with RVNet weights performs

the best as it correctly classifies examples of charcoal kilns while not making many false positive predictions.

Example predictions for burial mounds are shown in Figure 11. Similarly, ImageNet weights lead to the poorest results, many of the burial mounds are not detected. While RVNet and RVGAN weights produce better results, the model initialized with random weights gives the best predictions for this region.

The last region contains examples of mining holes as shown in Figure 12. Predictions by the model initialized with ImageNet weights are overestimated. There are big blobs of predictions which results into a lot of false positives. Predictions by the other models are better, and the predicted mining holes are well separated with better delineated outlines.

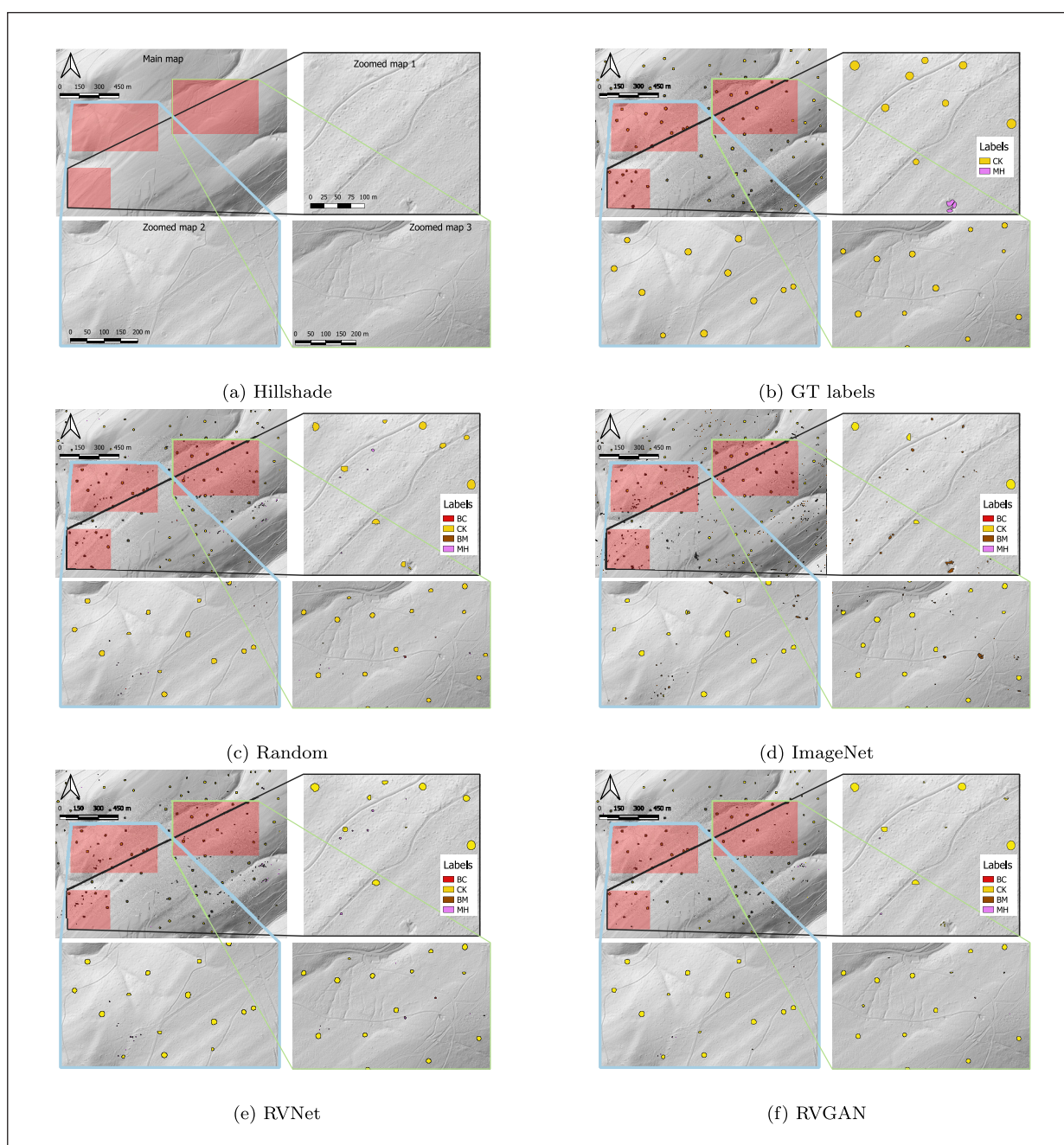


Figure 10 Example predictions for regions with Charcoal Kilns (CK).

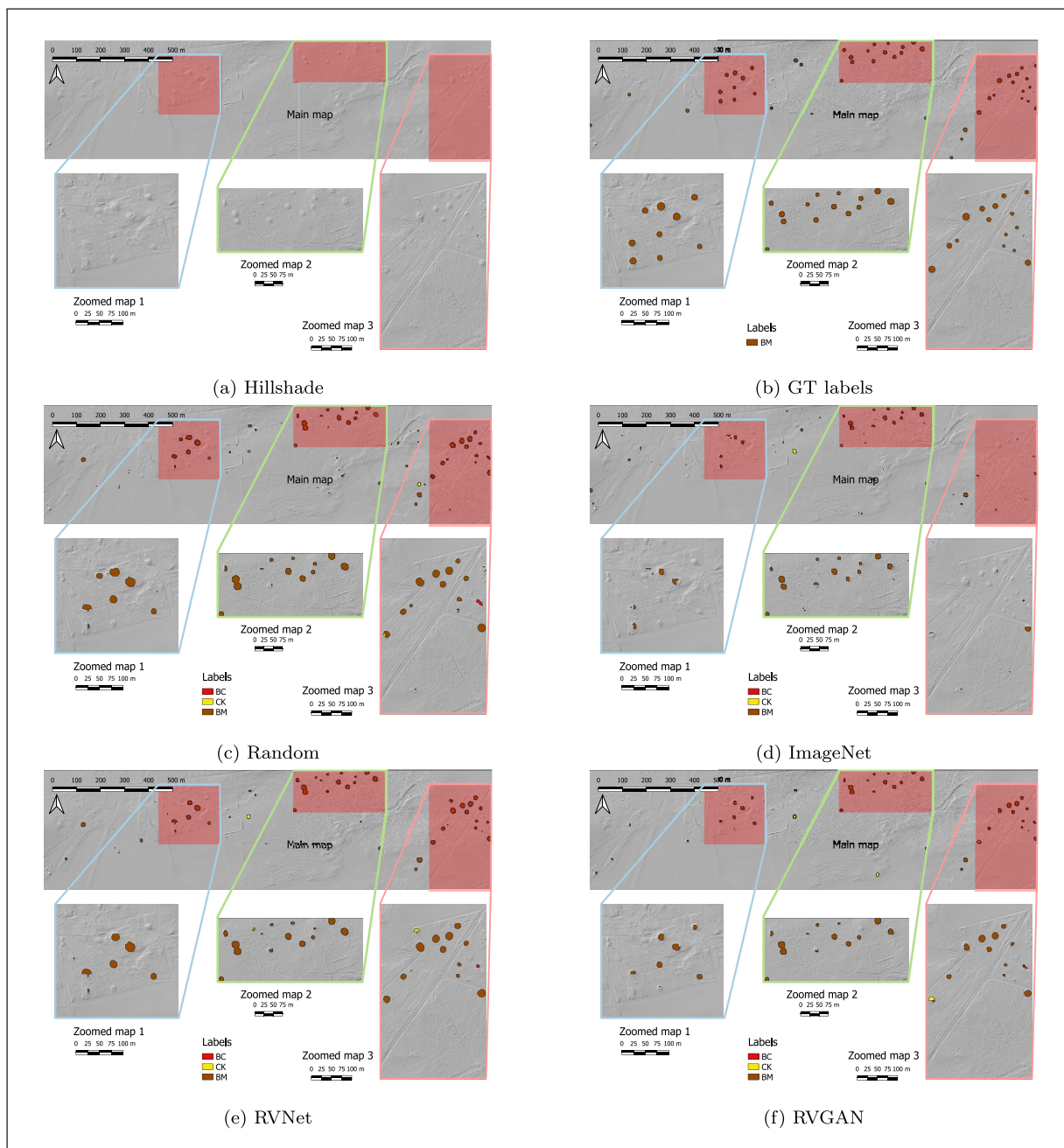


Figure 11 Example predictions for regions with Burial Mounds (BM).

5. CONCLUSION

In this research, Self-Supervised Learning (SSL) techniques were used for semantic segmentation of archaeological monuments in DTMs. Two models, RVNet and RVGAN, were pretrained on unlabeled DTM data and their learned weights were transferred to semantic segmentation. RVNet is an encoder-decoder architecture and RVGAN is a Pix2Pix-based Generative Adversarial Network (GAN) architecture. Both were trained to generate RVs such as LD, SLRM, Slope and SVF for a given DTM.

A small annotated DTM dataset was used to train a deep learning model for semantic segmentation of archaeological monuments such as bomb craters,

charcoal kilns, burial mounds and mining holes. Experiments show that the model initialized with pretrained RVNet and RVGAN weights outperforms the one with random weights. Moreover, using weights of models trained on natural images to initialize and fine-tune on the same DTM dataset results in lower performance compared to using weights from RVNet and RVGAN pretrained on DTM data. This is due to the difference in data domains.

The pretrained weights of RVNet and RVGAN and the implementations are made publicly available and we believe they can be used to fine-tune deep learning models on a variety of other supervised downstream tasks such as classification, object detection and instance segmentation in projects that use DTM data.

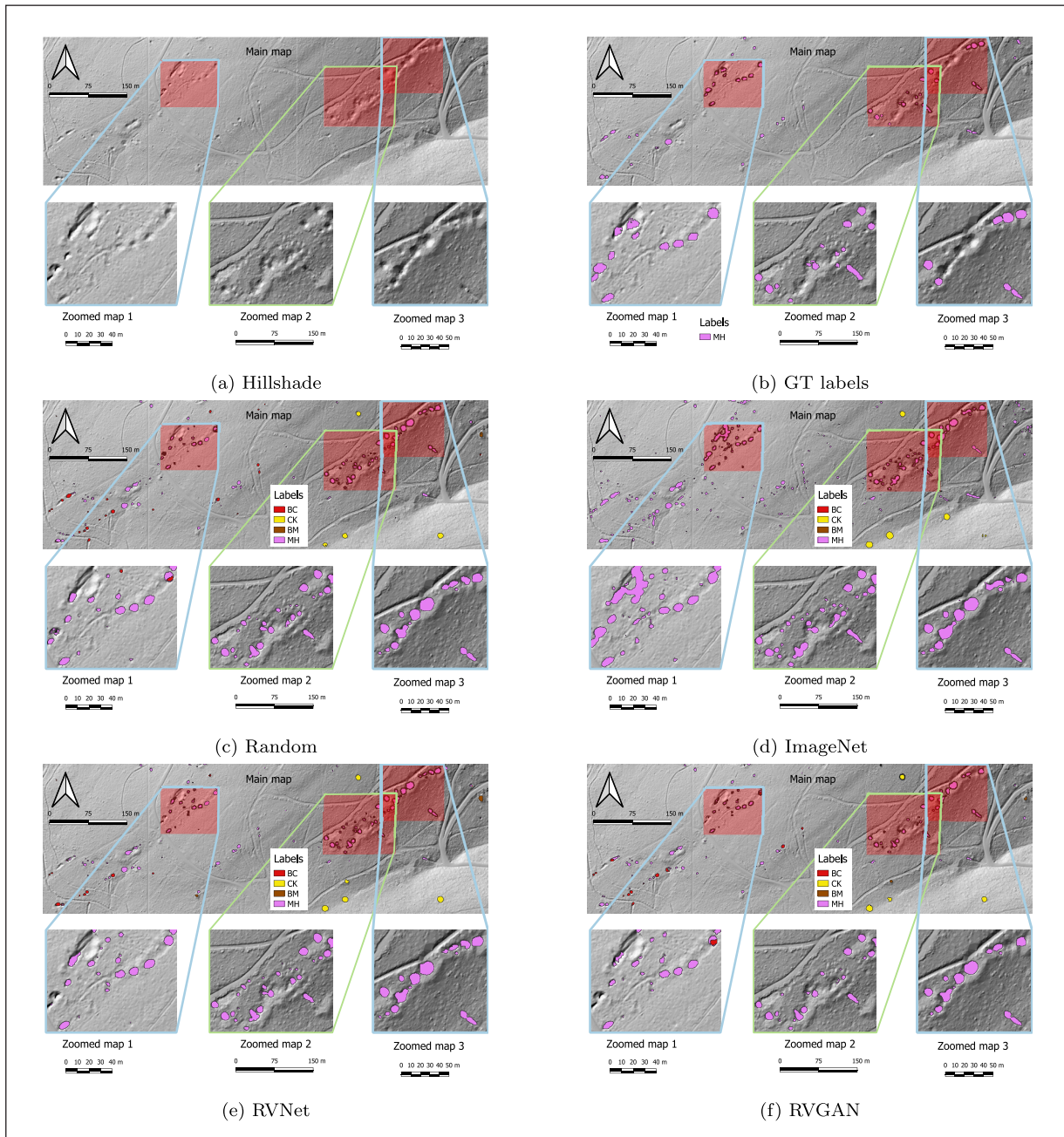


Figure 12 Example predictions for regions with Mining Holes (MH).

Due to the limited size of annotated dataset, future work in this direction includes exploring deep active learning. It is an iterative method where the model is first trained on a few annotated examples. The size of the training dataset is iteratively increased after each round, using the predictions of the model on unlabeled data and an acquisition function that determines which examples need to be annotated that could maximize the performance of the model.

Another self-supervised learning technique is called contrastive learning which aims to learn a good representation of the input data by contrasting different views of the same data. A model is trained to learn to distinguish between similar and dissimilar pairs of images and by doing so, it can learn a representation that

captures the underlying structure of the data (He et al., 2020; Grill et al., 2020). This is also a promising approach to be explored in the future as it can be trained on the DTM patches without the need for the corresponding RVs.

Additionally, after showing tremendous success in natural language processing, transformers (Wolf et al., 2020) have made their way into the computer vision community in the form of vision transformers (Khan et al., 2021). They provide some advantages over convolutional neural networks. Convolutions are sensitive to local patches and lack a global understanding of images while transformers are said to learn a better global representation of images. This study, i.e., self-supervised learning can be extended to shift from convolution to transformer-based models in the future.

NOTE

1 <https://github.com/SSL-DTM>.


FUNDING INFORMATION


This research was funded by the Lower Saxony Ministry of Science and Culture through the “Niedersächsisches Vorab” funding initiative and with the collaboration of Lower Saxony State Office for Heritage (Niedersächsisches Landesamt für Denkmalpflege). The publication of this article was funded by the Open Access Fund of the Leibniz Universität Hannover.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Bashir Kazimi  orcid.org/0000-0003-1802-7511
Leibniz University Hannover Institute of Cartography and
Geoinformatics Appelstr. 9a, 30167 Hannover, DE

Monika Sester  orcid.org/0000-0002-6656-8809
Leibniz University Hannover Institute of Cartography and
Geoinformatics Appelstr. 9a, 30167 Hannover, DE

REFERENCES

- Bonhage, A, Eltahir, M, Raab, T, Breuß, M, Raab, A** and **Schneider, A.** 2021. A modified mask region-based convolutional neural network approach for the automated detection of archaeological sites on high-resolution light detection and ranging-derived digital elevation models in the north german lowland. *Archaeological Prospection*, 28: 177–186. DOI: <https://doi.org/10.1002/arp.1806>
- Bundzel, M, Jaščur, M, Kováč, M, Lieskovský, T, Sinčák, P** and **Tkáčik, T.** 2020. Semantic segmentation of airborne lidar data in maya archaeology. *Remote Sensing* 12. URL: <https://www.mdpi.com/2072-4292/12/22/3685>. DOI: <https://doi.org/10.3390/rs12223685>
- Caesar, H, Uijlings, J** and **Ferrari, V.** 2018. Coco-stuff: Thing and stuff classes in context. In: *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, pp. 1209–1218. DOI: <https://doi.org/10.1109/CVPR.2018.00132>
- Caspari, G** and **Crespo, P.** 2019. Convolutional neural networks for archaeological site detection–finding “princely” tombs. *Journal of Archaeological Science*, 110: 104998. DOI: <https://doi.org/10.1016/j.jas.2019.104998>
- David, OE** and **Netanyahu, NS.** 2016. Deeppainter: Painter classification using deep convolutional autoencoders. In: *International Conference on Artificial Neural Networks*, Springer. pp. 20–28. DOI: https://doi.org/10.1007/978-3-319-44781-0_3
- Deng, J, Dong, W, Socher, R, Li, LJ, Li, K** and **Fei-Fei, L.** 2009. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE. pp. 248–255. DOI: <https://doi.org/10.1109/CVPR.2009.5206848>
- Doersch, C, Gupta, A** and **Efros, AA.** 2015. Unsupervised visual representation learning by context prediction. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430. DOI: <https://doi.org/10.1109/ICCV.2015.167>
- Du, L, You, X, Li, K, Meng, L, Cheng, G, Xiong, L** and **Wang, G.** 2019. Multimodal deep learning for landform recognition. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158: 63–75. DOI: <https://doi.org/10.1016/j.isprsjprs.2019.09.018>
- Everingham, M, Eslami, SMA, Van Gool, L, Williams, CKI, Winn, J** and **Zisserman, A.** 2015. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111: 98–136. DOI: <https://doi.org/10.1007/s11263-014-0733-5>
- Gelbman, E** and **Papo, H.** 1984. Digital terrain models for slopes and curvatures. *Photogrammetric Engineering and Remote Sensing*, 50: 695–701.
- Gidaris, S, Singh, P** and **Komodakis, N.** 2018. Unsupervised representation learning by predicting image rotations. In: *International Conference on Learning Representations*.
- Goodfellow, I, Pouget-Abadie, J, Mirza, M, Xu, B, Warde-Farley, D, Ozair, S, Courville, A** and **Bengio, Y.** 2014. Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, pp. 2672–2680.
- Grill, JB, Strub, F, Altché, F, Tallec, C, Richemond, P, Buchatskaya, E, Doersch, C, Avila Pires, B, Guo, Z, Gheslaghi Azar, M,** et al. 2020. Bootstrap your own latent—a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33: 21271–21284.
- Guo, X, Liu, X, Zhu, E** and **Yin, J.** 2017. Deep clustering with convolutional autoencoders. In: *International Conference on Neural Information Processing*, Springer. pp. 373–382. DOI: https://doi.org/10.1007/978-3-319-70096-0_39
- Guyot, A, Hubert-Moy, L** and **Lorho, T.** 2018. Detecting neolithic burial mounds from lidar-derived elevation data using a multi-scale approach and machine learning techniques. *Remote Sensing*, 10: 225. DOI: <https://doi.org/10.3390/rs10020225>
- He, K, Fan, H, Wu, Y, Xie, S** and **Girshick, R.** 2020. Momentum contrast for unsupervised visual representation learning. In: *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, pp. 9729–9738. DOI: <https://doi.org/10.1109/CVPR42600.2020.00975>
- He, K, Gkioxari, G, Dollár, P** and **Girshick, R.** 2017. Mask r-cnn. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969. DOI: <https://doi.org/10.1109/ICCV.2017.322>

- Hesse, R.** 2016. Visualisierung hochauflösender digitaler geländemodelle mit livt.
- Isola, P, Zhu, JY, Zhou, T and Efros, AA.** 2017. Image-to-image translation with conditional adversarial networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134. DOI: <https://doi.org/10.1109/CVPR.2017.632>
- Jin, Y, Zhang, J, Li, M, Tian, Y and Zhu, H.** 2017. Towards the high-quality anime characters generation with generative adversarial networks. In: *Proceedings of the Machine Learning for Creativity and Design Workshop at Neural Information Processing Systems*.
- Kazimi, B, Malek, K, Thiemann, F and Sester, M.** 2020a. Semi supervised learning for archaeological object detection in digital terrain models. In: *International Conference on Cultural Heritage and New Technologies 2020*.
- Kazimi, B, Thiemann, F, Malek, K, Sester, M and Khoshelham, K.** 2018. Deep learning for archaeological object detection in airborne laser scanning data. In: *Proceedings of the 2nd Workshop On Computing Techniques For Spatio-Temporal Data in Archaeology And Cultural Heritage co-located with 10th International Conference on Geographical Information Science*.
- Kazimi, B, Thiemann, F and Sester, M.** 2019. Object instance segmentation in digital terrain models. In: Vento, M and Percannella, G (Eds.), *Computer Analysis of Images and Patterns, Springer International Publishing*, Cham. pp. 488–495. DOI: https://doi.org/10.1007/978-3-030-29891-3_43
- Kazimi, B, Thiemann, F and Sester, M.** 2020b. Detection of terrain structures in airborne laser scanning data using deep learning. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 5.
- Khan, S, Naseer, M, Hayat, M, Zamir, SW, Khan, FS and Shah, M.** 2021. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*. DOI: <https://doi.org/10.1145/3505244>
- Kingma, DP and Ba, J.** 2015. Adam: a method for stochastic optimization. In: Bengio, Y and LeCun, Y (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL: <http://arxiv.org/abs/1412.6980>.
- Kokalj, Ž and Hesse, R.** 2017. Airborne laser scanning raster data visualization: A Guide to Good Practice. DOI: <https://doi.org/10.3986/9789612549848>
- Kokalj, Ž and Somrak, M.** 2019. Why not a single image? combining visualizations to facilitate fieldwork and on-screen mapping. *Remote Sensing*, 11: 747. DOI: <https://doi.org/10.3390/rs11070747>
- Krishna, ST and Kalluri, HK.** 2019. Deep learning and transfer learning approaches for image classification. *International Journal of Recent Technology and Engineering (IJRTE)*, 7: 427–432.
- Krizhevsky, A, Sutskever, I and Hinton, GE.** 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25.
- Kumar, E and Bindu, C.** 2019. Medical image analysis using deep learning: a systematic literature review. In: *International Conference on Emerging Technologies in Computer Engineering*, Springer. pp. 81–97. DOI: https://doi.org/10.1007/978-981-13-8300-7_8
- Ledig, C, Theis, L, Huszár, F, Caballero, J, Cunningham, A, Acosta, A, Aitken, A, Tejani, A, Totz, J, Wang, Z, et al.** 2017. Photo-realistic single image superresolution using a generative adversarial network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4681–4690. DOI: <https://doi.org/10.1109/CVPR.2017.19>
- Masci, J, Meier, U, Ciresan, D and Schmidhuber, J.** 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In: *International Conference on Artificial Neural Networks*, Springer. pp. 52–59. DOI: https://doi.org/10.1007/978-3-642-21735-7_7
- Maxwell, AE, Pourmohammadi, P and Poyner, JD.** 2020. Mapping the topographic features of mining-related valley fills using mask r-cnn deep learning and digital elevation data. *Remote Sensing*, 12: 547. DOI: <https://doi.org/10.3390/rs12030547>
- Noroozi, M and Favaro, P.** 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In: *European Conference on Computer Vision*, Springer. pp. 69–84. DOI: https://doi.org/10.1007/978-3-319-46466-4_5
- Øivind Due Trier, Reksten, JH and Løseth, K.** 2021. Automated mapping of cultural heritage in norway from airborne lidar data using faster rcnn. *International Journal of Applied Earth Observation and Geoinformation*, 95: 102241. URL: <https://www.sciencedirect.com/science/article/pii/S0303243420308849>. DOI: <https://doi.org/10.1016/j.jag.2020.102241>
- Paszke, A, Gross, S, Massa, F, Lerer, A, Bradbury, J, Chanan, G, Killeen, T, Lin, Z, Gimelshein, N, Antiga, L, Desmaison, A, Kopf, A, Yang, E, DeVito, Z, Raison, M, Tejani, A, Chilamkurthy, S, Steiner, B, Fang, L, Bai, J and Chintala, S.** 2019. Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H, Larochelle, H, Beygelzimer, A, d'Alché-Buc, F, Fox, E and Garnett, R (Eds.), *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 8024–8035.
- Pathak, D, Krahenbuhl, P, Donahue, J, Darrell, T and Efros, AA.** 2016. Context encoders: Feature learning by inpainting. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544. DOI: <https://doi.org/10.1109/CVPR.2016.278>
- Politz, F, Kazimi, B and Sester, M.** 2018. Classification of laser scanning data using deep learning. 38th Scientific Technical Annual Meeting of the German Society for Photogrammetry, Remote Sensing and Geoinformation 27. URL: <https://pdfs.semanticscholar.org/698c/924265e469d58eb6ffd7e561c2d2b4814a06.pdf>.
- Ren, S, He, K, Girshick, R and Sun, J.** 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28.
- Satari, R, Kazimi, B and Sester, M.** 2021. Extraction of linear structures from digital terrain models using deep

- learning. *AGILE: GIScience Series*, 2: 1–14. DOI: <https://doi.org/10.5194/agile-giss-2-11-2021>
- Somrak, M, Džeroski, S and Kokalj, Ž.** 2020. Learning to classify structures in alsderived visualizations of ancient maya settlements with cnn. *Remote Sensing*, 12: 2215. DOI: <https://doi.org/10.3390/rs12142215>
- Soroush, M, Mehrtash, A, Khazraee, E and Ur, JA.** 2020. Deep learning in archaeological remote sensing: Automated qanat detection in the kurdistan region of Iraq. *Remote Sensing*, 12: 500. DOI: <https://doi.org/10.3390/rs12030500>
- Stahlberg, F.** 2020. Neural machine translation: A review. *Journal of Artificial Intelligence Research*, 69: 343–418. DOI: <https://doi.org/10.1613/jair.1.12007>
- Sun, K, Xiao, B, Liu, D and Wang, J.** 2019. Deep high-resolution representation learning for human pose estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5693–5703. DOI: <https://doi.org/10.1109/CVPR.2019.00584>
- Trier, ØD, Cowley, DC and Waldeland, AU.** 2019. Using deep neural networks on airborne laser scanning data: Results from a case study of semi-automatic mapping of archaeological topography on Arran, Scotland. *Archaeological Prospection*, 26, 165–175. DOI: <https://doi.org/10.1002/arp.1731>
- Trier, ØD, Reksten, JH and Løseth, K.** 2021. Automated mapping of cultural heritage in norway from airborne lidar data using faster r-cnn. *International Journal of Applied Earth Observation and Geoinformation*, 95: 102241. DOI: <https://doi.org/10.1016/j.jag.2020.102241>
- Trier, ØD, Salberg, AB and Pilø, LH.** 2018. Semi-automatic mapping of charcoal kilns from airborne laser scanning data using deep learning. In: *CAA2016: Oceans of Data. Proceedings of the 44th Conference on Computer Applications and Quantitative Methods in Archaeology*, Archaeopress Oxford. pp. 219–231.
- Verschoof-van der Vaart, WB and Lambers, K.** 2019. Learning to look at lidar: The use of r-cnn in the automated detection of archaeological objects in lidar data from the Netherlands. *Journal of Computer Applications in Archaeology*, 2. DOI: <https://doi.org/10.5334/jcaa.32>
- Vondrick, C, Pirsivash, H and Torralba, A.** 2016. Generating videos with scene dynamics. In: *Advances in Neural Information Processing Systems*, pp. 613–621.
- Voulodimos, A, Doulamis, N, Doulamis, A and Protopapadakis, E.** 2018. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience 2018*. DOI: <https://doi.org/10.1155/2018/7068349>
- Wolf, T, Debut, L, Sanh, V, Chaumond, J, Delangue, C, Moi, A, Cistac, P, Rault, T, Louf, R, Funtowicz, M, Davison, J, Shleifer, S, von Platen, P, Ma, C, Jernite, Y, Plu, J, Xu, C, Scao, TL, Gugger, S, Drame, M, Lhoest, Q and Rush, AM.** 2020. Transformers: State-of-the-art natural language processing. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Online. pp. 38–45. URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>. DOI: <https://doi.org/10.18653/v1/2020.emnlp-demos.6>
- Wu, H, Zheng, S, Zhang, J and Huang, K.** 2019. Gp-gan: Towards realistic highresolution image blending. In: *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2487–2495. DOI: <https://doi.org/10.1145/3343031.3350944>
- Zakšek, K, Oštir, K and Kokalj, Ž.** 2011. Sky-view factor as a relief visualization technique. *Remote sensing*, 3: 398–415. DOI: <https://doi.org/10.3390/rs3020398>
- Zhang, H, Xu, T, Li, H, Zhang, S, Wang, X, Huang, X and Metaxas, DN.** 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5907–5915. DOI: <https://doi.org/10.1109/ICCV.2017.629>
- Zhang, R, Isola, P and Efros, AA.** 2016. Colorful image colorization. In: *European Conference on Computer Vision*, Springer. pp. 649–666. DOI: https://doi.org/10.1007/978-3-319-46487-9_40
- Zhao, ZQ, Zheng, P, Xu, ST and Wu, X.** 2019. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30: 3212–3232. DOI: <https://doi.org/10.1109/TNNLS.2018.2876865>
- Zhou, B, Zhao, H, Puig, X, Fidler, S, Barriuso, A and Torralba, A.** 2017. Scene parsing through ade20k dataset. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 633–641. DOI: <https://doi.org/10.1109/CVPR.2017.544>
- Zhu, JY, Park, T, Isola, P and Efros, AA.** 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232. DOI: <https://doi.org/10.1109/ICCV.2017.244>

TO CITE THIS ARTICLE:

Kazimi, B and Sester, M. 2023. Self-Supervised Learning for Semantic Segmentation of Archaeological Monuments in DTMs. *Journal of Computer Applications in Archaeology*, 6(1): 155–173. DOI: <https://doi.org/10.5334/jcaa.110>

Submitted: 02 April 2023 **Accepted:** 16 October 2023 **Published:** 23 November 2023

COPYRIGHT:

© 2023 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Computer Applications in Archaeology is a peer-reviewed open access journal published by Ubiquity Press.

