Terrain-Informed Self-Supervised Learning: Enhancing Building Footprint Extraction from LiDAR Data with Limited Annotations

Anuja Vats, David Völgyes, Martijn Vermeer, Marius Pedersen, Kiran Raja, Daniele S. M. Fantin and Jacob Alexander Hay^{*}

April 19, 2024

Abstract

Estimating building footprint maps from geospatial data is vital in urban planning, development, disaster management, and various other applications. Deep learning methodologies have gained prominence in building segmentation maps, offering the promise of precise footprint extraction without extensive post-processing. However, these methods face challenges in generalization and label efficiency, particularly in remote sensing, where obtaining accurate labels can be both expensive and time-consuming. To address these challenges, we propose terrain-aware self-supervised learning, tailored to remote sensing, using digital elevation models from LIght Detection and Ranging (LiDAR) data. We propose to learn a model to differentiate between bare Earth and superimposed structures enabling the network to implicitly learn domain-relevant features without the need for extensive pixel-level annotations. We test the effectiveness of our approach by evaluating building segmentation performance on test datasets with varying label fractions. Remarkably, with only 1% of the labels (equivalent to 25 labeled examples), our method improves over ImageNet pretraining, showing the advantage of leveraging unlabeled data for feature extraction in the domain of remote sensing. The performance improvement is more pronounced in few-shot scenarios and gradually closes the gap with ImageNet pretraining as the label fraction increases. We test on a dataset characterized by substantial distribution shifts (including resolution variation and labeling errors) to demonstrate the generalizability of our approach. When compared to other baselines, including ImageNet pretraining and more complex architectures, our approach consistently performs better, demonstrating the efficiency and effectiveness of self-supervised terrainaware feature learning.

Index Terms— remote sensing, self-supervised learning, LiDAR, building, segmentation

1 Introduction

Estimating building footprint maps from geospatial data has been seen as indispensable for activities pertaining to urban planning and development such as estimating population distribution, monitoring urban expansion and impervious areas, creating detailed 3D city models, transportation, and navigation, as well as detecting illegal construction cases. In scenarios where timely information is crucial, such as assessing damage in the aftermath of natural disasters or updating topographical databases at a national scale, methods for creating accurate and updated maps of dynamic geographies are paramount.

Recent methods for building segmentation have shifted towards the application of deep learning methodologies [1], offering the prospect of more precise building footprint extraction while minimizing the need for extensive post-processing. The segmentation task is difficult due to the vast heterogeneity across urban scenes globally. Factors such as variabilities across sensors, data quality, resolution, atmospheric and topographic differences, seasonal variations, etc. have been very challenging to counter when generalizing to unseen geographies [2]. Further, the most noteworthy results in supervised learning in remote sensing stem from learning on a large number of accurately labeled images. Although open mapping platforms such as OpenStreetMap (OSM) can be used for acquiring labels, they may not be ideal due to inconsistencies between the vector data from the platform and the acquired data. Misalignments may arise due to differences in spatial resolution or time differences resulting in topography changes. The accuracy of labels derived from resolving for resolution differences can result in errors (such as those highlighted in later sections). Moreover, as open platforms rely on contributions from individuals on a voluntary basis, the label density varies around the globe, where certain high-resource regions such as Europe have more labels than regions such as Africa or Asia [3]. This may result in incomplete or outdated data.

For certain specialized remote sensing applications, the predefined categories in platforms like OSM may even be too general for use, requiring experts to annotate the data effectively. This is not only expensive but typically requires manual work by a domain expert [4, 5]. In most real-world scenarios there is a shortage of labels at the desired level of preciseness. Segmentation tasks face a particular challenge exacerbated by the time-intensive nature of pixel-level labeling. Furthermore, in contrast to certain domains where data scarcity is a primary

^{*}Anuja Vats, Marius Pedersen and Kiran Raja are with the Department of Computer Science, Norwegian Institute of Science and Technology, Gjøvik, Norway (e-mail: anuja.vats@ntnu.no, marius.pedersen@ntnu.no, kiran.raja@ntnu.no).

[†]David Völgyes, Martijn Vermeer, Daniele S. M. Fantin and Jacob Alexander Hay are with Science and Technology AS, Tordenskioldsgate 2, 0160 Oslo, Norway (e-mail: volgyes@stcorp.no, vermeer@stcorp.no, fantin@stcorp.no, hay@stcorp.no)

concern, remote sensing offers a unique advantage: extensive datasets collected over time, covering diverse geographic areas and conditions, sourced from satellites orbiting the Earth. This can be taken advantage of using alternative methods of acquiring supervision, particularly from unlabeled datasets. Recent studies have shown the value of leveraging the intrinsic structure and patterns inherent within unlabeled data. This learning paradigm, known as self-supervised learning derives supervisory signals from data through pretext tasks. Learning in this way enables discovering domain-relevant features that can greatly improve segmentation tasks of interest while requiring fewer pixel-level annotations [6, 7].

This work targets the two aforementioned challenges that persist in the domain of remote sensing for improvement (i) generalization and (ii) label-efficiency. We propose a terrainaware pretext task for learning features that can generalize despite commonly encountered domain shifts such as in input resolutions, geographies, and labeling errors (generalization) and perform well on downstream building segmentation with only a few labels (label efficiency). Further, although the core motivation behind our design is to improve generalization performance while strictly constraining the annotation requirement, we also show that a focus on constructing a more efficient learning strategy allows simplifying computational complexity, where already available simple models like U-Net [8] perform on par with computationally much more complex models such as transformers [9, 10] promoting efficiency and resource conservation. While the use of segmentation models based on complex architectures such as transformers has become mainstream [10], in this paper, we argue in favour of investigating if the same advantages may be obtained at lower computation complexity, by improving learning strategies. To this end, we perform self-supervised learning instead of supervised learning, to learn generalizable features without labels. Since a self-supervised task such as ours is not learning via class-discrimination (as in many classification tasks), it can learn features that tend to be more informative of the input domain as opposed to discriminative. Such informative features lend to easy adaptation in the face of generalization, not just to out-of-distribution samples later on, but also to various tasks such as tree-species classification, land-use segmentation, etc. The contributions of this work are:

- We propose a new pretext task for learning generalizable features from unlabeled LiDAR data catering specifically to the domain of remote sensing considering the limited-label availability scenario.
- Our approach demonstrates enhanced performance compared to supervised ImageNet pretraining, resulting in a substantial reduction in the number of labels required for downstream training. Table 1 displays the results in a few-shot learning setting, where we observe improved performance over ImageNet pretraining even with just 25 examples (equivalent to 1% of the labels for fine-tuning). This indicates the efficacy of our terrain-aware pretext task in minimizing label requirements in the field of remote sensing.
- Our method also shows better generalization performance under radical domain shift between training and test

datasets. We surpass the baseline model (U-Net Random) by a margin of 0.32 in Intersection over Union (IoU) (0.844 vs 0.521 Intersection over Union in Table 2 indicating the benefit from our learning strategy in feature reuse and transfer) and a transformer-based model by a margin of 0.054 in IoU (0.844 vs 0.790 indicating gain in computational complexity).

In this study, our focus shifts away from spectral data, and instead, we use the Norwegian national elevation model derived from aerial LiDAR data¹. Specifically, we employ three distinct Digital Elevation Models (DEMs). These DEMs capture and characterize Earth's terrain in a three-dimensional context. A Digital Surface Model (DSM), represents the uppermost surface of the Earth's landscape. It encompasses all objects and terrain features, including natural elements like trees, man-made structures such as buildings, and other terrain irregularities. Essentially, it is a representation of the Earth's surface, including both ground and above-ground elements. A Digital Terrain Model (DTM) on the other hand, specifically isolates the bare Earth's topography, excluding any aboveground objects such as buildings and vegetation. It provides a representation of the ground surface free from the influence of man-made features or vegetation. Thus serving as a baseline elevation model, aiding in the identification of landforms and contours. A third model, known as the normalized Digital Surface Model (nDSM) (alternatively referred to as Canopy Height Model (CHM) in forestry applications when dealing with the canopy of the forest) is derived by subtracting the DTM from the DSM. This subtraction isolates the height of structures above ground enabling knowledge of vertical structures and the density of objects within a given area. These models (DSM, DTM, nDSM) are gridded (alternatively referred to as rasterizing), georeferenced, and 2D projected into local geospatial coordinate reference systems. The 3D LiDAR point cloud can be processed into DTM and DSM with traditional methods [11, 12]. Figure 1 shows the three different elevation datasets derived from the LiDAR data.

Digital Surface Model

Canopy Height Model

Figure 1: Different digital elevation models characterizing different aspects of the Earth's surface derived from LiDAR point clouds.

Digital Terrain Model

2 Related work

Building segmentation

Owing to its importance in urban planning, several approaches have been developed [13–17] with the aim of continuously

¹https://hoydedata.no/

improving the detection of buildings across different landscapes. The authors in [13, 16] use high-resolution images for improving building delineation. To overcome challenges associated with building displacements and shadows, integrating data from multiple sources, such as LiDAR in addition to spectral information from aerial and satellite images has been shown to enhance the robustness and accuracy of building segmentation [14, 18–20]. Some approaches have also utilized stereo images [21] to create stereo point clouds which in turn can be used to create surface maps. However, despite advantages [22], relatively fewer approaches depend solely on Li-DAR data for segmentation. This is due to difficulties associated primarily with identifying and eliminating vegetation from LiDAR data as well as constraints related to cost and availability. Despite this, it is valuable to exploit the rich 3D structural information available from LiDAR data for building segmentation, independently from spectral information as fusing sources not only leaves room for introducing registration and resampling errors but allows developing potentially unnecessary reliance on an additional source during inference. In this work, we show that gridded LiDAR elevation data solely, without requiring elaborate pre-processing techniques, can lead to comparable or better performance on building segmentation as compared to methods that combine LiDAR data with alternative sensor information.

Self-supervised learning

Self-supervised learning (SSL) is learning by encoding unlabeled data to a lower-dimensional representation while being constrained to preserve information beneficial for downstream tasks such as classification. This step is often known as self-supervised pretraining. The supervisory signal during pretraining is derived from a surrogate/pretext task designed to extract desirable information (desirable to downstream tasks, while ignoring other task-irrelevant features) implicit within the data. A common approach is to use the network to predict the structural or semantic properties of the data. By learning to recover this information, features regarding color, texture, scale, etc. may be extracted. The encoded features are then transferred to (downstream) tasks of interest such as building or road segmentation, land use classification, etc. Learning in this way has been shown to perform better than supervised pretraining on many problems [23-25]. In remote sensing, Zhang *et al.* [26] perform self-supervised rotation-angle prediction along with target recognition within a multi-task learning framework, where the bottom layers share parameters between the two tasks. Similarly, Zhao et al. [27] combine rotation prediction with scene classification, such that weight parameters for each loss are sampled randomly from a beta distribution. Tao *et al.* [28] analyze the impact of learning via different pretext tasks, these tasks being image inpainting, predicting relative position between image patches, and instance discrimination. Vincenzi et al. [29] employ a colorization task for self-supervision, where the networks predict RGB channel information from other spectral bands. They emphasize the importance of tailoring pretext tasks specifically to remote sensing rather than translating directly from other domains to extract beneficial task-specific features. Further, typical selfsupervision approaches like colorization, seasonal contrast or rotation prediction are not designed for remote sensing and thus are not applicable directly to LiDAR data. For example, data from different seasons may not be available for contrasting or rotated versions of LiDAR images may appear as natural as original images (as opposed to rotated image of a tree). A self-supervised task such as colorization or inpainting would necessitate an aligned secondary source additionally to LiDAR. Unlike them, we propose a pretext task for elevation/LiDAR data catering specifically to remote sensing for learning effective and generalizable features from LiDAR data in a label-free manner.

3 Methodology

As discussed prior, self-supervision is an effective way to reduce the amount of training data required for a task by pretraining a model on a dummy pretext task $T_{pretext}$, labeled data for which can be freely curated, $(X, Y_{pretext})$. The selfsupervised learning objective can be defined as minimizing the discrepancy between predicted labels $Y'_{pretext}$ and artificially generated $Y_{pretext}$.

$$min_{\theta} = L_{pretext}(Y_{pretext}, Y_{pretext}) \tag{1}$$

In this work, we propose a pretext task $T_{pretext}$ to predict DTM (terrain) from DSM (surface). Essentially, the task is to remove all the above-ground structures such as vegetation, buildings, houses, powerlines, etc, to recover the underlying topography encompassing components of natural landscapes. However, since the effectiveness of this strategy depends on the extent to which the acquired representation from the pretext task $\phi(X)$ can be transferred and minimally adapted for use in the downstream task, the choice of the task should be such as to encourage a unified and transferable feature space $\phi(X)$ that can be effectively utilized for target task of interest.

Considering this requirement of pretext tasks, our $T_{pretext}$ forces the network to differentiate between which elevation values correspond to bare-Earth as opposed to those of superimposed structures, such as vegetation and buildings. Thus encouraging the network to establish relationships between the various types of structures and the underlying terrain. The relationships could be the presence of fewer artificial objects on mountainous terrain compared to urban terrain, discernible variations in tree heights across diverse terrains, and fluctuations in building size, shape, and typology relative to the nature of the terrain. Our pretext task facilitates learning these relationships among different objects and their interplay with the terrain, without the need for explicit labels.

Further, considering that segmentation networks struggle when it comes to identifying buildings that exhibit diverse shapes and scales [30], it is advantageous to have a $T_{pretext}$ that counters this. Also, the quality of labels plays a role as the DTM/DSM used for deriving segmentation labels may be inaccurate, further affecting learning. Refer to Figure 2 which visually demonstrates the size and scale differences that we work with in our datasets. While several approaches have been developed to mitigate resolution differences between training and testing scenarios [31–33], a degree of invariance to scale is implicit in our $T_{pretext}$. This is because the expected scale of objects may be inferred indirectly based on the characteristics of the terrain itself.

The learned $\phi(X)$ is then fine-tuned on the downstream task with explicitly acquired labels (Y_{down}) as shown in Eq. 2

$$min_{\theta} = L_{down}(Y_{down}, Y'_{down}; \phi(X); \theta)$$
(2)

where Y'_{down} are the predicted labels and θ represents the downstream model parameters.



Figure 2: Scale differences as seen in our datasets. The top row shows LiDAR images at different scales from the train (left) and test (right) datasets. The corresponding building labels are shown in the bottom row. Even though our approach is pretrained for terrain-recovery on images as seen on the left, it translates easily to building with scales and sizes as seen on the right, indicating generalizability to objects at different scales (The magnification is to enable visual comparison between the two scales and aid reader comprehension and is not drawn to scale).

3.1 Pretraining

We use U-Net [8] with a ResNet-50 [34] encoder for the image reconstruction pretext task. The network takes an object raster (x_0) as input and predicts the corresponding terrain image (x_t) as seen in Figure 3. We use an image reconstruction task, as this has the advantage of allowing us to reuse all the layers within the encoder as well as the decoder for our downstream task of segmentation. This is in contrast to with how transfer happens for a classification pretext task [27, 35], where only the encoder weights would be useful. The outputs (logits) from the final convolutional layer are used to evaluate the reconstruction loss. We use a combination of two losses to evaluate the reconstruction, the first being a smooth-L1 loss [36] and the second being LPIPS (learned perceptual image patch similarity) loss [35]. The smooth-L1 (Eq. 3) loss converges to either L1 or L2-like loss depending on the hyperparameter β chosen as 1.0 in this experiment. This allows the network to be more forgiving to outliers that L2 loss alone would penalize heavily, while benefiting from L1-like slightly faster convergence.

$$L_{\text{smooth L1}}(y, y') = \begin{cases} \frac{0.5(y-y')^2}{\beta} & \text{if } |y-y'| < \beta\\ |y-y'| - 0.5\beta & \text{otherwise} \end{cases}$$
(3)

The LPIPS loss (Eq. 4) on the other hand, does not measure pixel-wise similarity between images, but instead measures the

perceptual similarity in the target x_t and reconstructed x_t^* terrain images. It does so by computing Euclidean distances between the two images in the feature space of pretrained convolutions networks \mathcal{F} such as VGG [37] or AlexNet [38].

$$LPIPS(x_t, x_t^*) = \sum_{l \in \mathcal{F}} w_l \cdot \mathsf{MSE}(\phi_l(x_t), \phi_l(x_t^*))$$
(4)

Here, $\phi_l(\cdot)$ refers to the normalized output of layer l within VGG network \mathcal{F} . w_l are the weights for the linear combination of the extracted features and are set to 1 in our case. Further, while the use of traditional metrics such as SSIM (Structural Similarity Index) [39] and MS-SSIM (Multi-Scale Structural Similarity Index) [40] are very common as loss functions for a reconstruction task such as this, we found that the negligible advantage obtained was negated by the computation overhead especially when evaluating similarity between the image and its reconstruction at multiple-scales (MS-SSIM). On the other hand, while LPIPS is associated with percepted smoothness, the high perceptual quality is actually a consequence of minimizing the distances between the images at different levels in the feature space. Although perceptual quality is not the primary concern in our case, we trade the computational complexity of MS-SSIM for LPIPS for slightly better reconstructions.

In addition, we use squeeze-and-excitation (SE) attention blocks [41] to capture channel-wise dependencies within the feature maps. This can be used to adaptively emphasize some channels while suppressing other less informative ones. SE blocks perform this in two steps via "squeeze" and "excitation". The squeeze step uses a global average pooling layer to compress each channel into a scalar descriptor representing the importance of that channel. The channel-wise descriptors are then allowed to interact through fully connected layers to combine and adaptively weight them based on importance. This is the excitation step. The weights can then be applied to original features to rescale them. This has been shown to allow networks the ability to selectively choose among the different feature maps adaptively, which improves segmentation [41].

4 Evaluation and Results

4.1 Dataset Details

4.1.1 Train dataset D1

Our primary data source consists of DTM and DSM models for Norway at 1m ground resolution in the UTM 33N projection. This data is published by the Norwegian Mapping Authority (in Norwegian: Kartverket), and is freely available on their website¹, under CC BY 4.0 license. The data is collected as 3D LiDAR point clouds via periodic aerial campaigns, where the maximum offset from the nadir is 20 degrees. Both point clouds and the gridded, reprojected DTM and DSM models are available. In this paper, only the processed DTM and DSM models are used. The subdataset used in this paper is republished under CC BY 4.0 license ensuring reproducibility.

To acquire labels for the same region, the Open-StreetMap²(OSM) database has been used from which build-

²https://openstreetmap.org/copyright



Figure 3: Our approach: We employ U-Net with Resnet-50 encoder for image reconstruction task from LiDAR surface image to terrain image. Through formulating the pretraining as a reconstruction task, all the layers within the encoder and decoder (except for the last task-specific layer) can be utilized in the downstream tasks as shown. *Conv* stands for convolutional operation, *BN* stands for batch normalization.



Figure 4: Training dataset (D1) and testing datasets (T1 and T2): During pretraining the model learns to reconstruct DTM from DSM at 1 m ground resolution. Both testing datasets T1 and T2 input nDSM images for building segmentation, introducing a shift from training data. Further, T2 additionally introduces resolution shift and label noise for testing the generalizability of our approach. Some of the labeling errors can be seen in the figure where added building refers to buildings that are found in the label but are absent in the LiDAR images.

ing footprints were rasterized to the same grid as the DTM and DSM models. These rasterized labels are published under the Open Database License (ODbL) in accordance with the OSM license requirements.

4.1.2 Test Datasets

For testing, we use two different datasets, Norway data T1 acquired in the same way as D1, and dataset T2 from NORA's MapAI competition [42]. Compared to the training dataset, T2 reflects acquisition and distortion shifts including but not limited to varying quality and different types of label noise (incomplete and incorrect labels, shown in Figures 4 and 6). Note that the MapAI dataset's exact resolution is unknown, and the data is not georeferenced. In this work, we use only a subset of the total challenge dataset (LiDAR images and mask for task 2 in the challenge, the accompanying spectral images have not been used as our work focuses on building detection from Li-DAR data only) as T2. The train and test dataset curated for this work can be accessed *here*.

This section discusses the evaluation protocol and results from the terrain-aware pretraining. We use two metrics as indicators of performance given by IoU (Intersection over Union) and bIoU (boundary IoU). IoU is defined as the ratio of the intersection of two sets to their union (Eq. 5). In segmentation, the two sets are the sets of pixels that are marked as the predicted object, Pred, and the ground truth, GT, respectively.

$$IoU = \frac{|Pred \cap GT|}{|Pred \cup GT|} \tag{5}$$

bIoU extends the metric to evaluate the boundaries (external edge of the buildings) more specifically and is thus IoU over masks considering a specified pixel thickness d along the boundaries.

4.2 Details on Pretraining and Training

The pretraining dataset consists of 16323 images for training and 1813 for validation with a patch size of 512x512 pixels. We use the Adam optimizer with an initial learning rate of 1×10^{-6} and weight decay of 1×10^{-8} at a batch size of 8. The learning rate is reduced by a factor of 0.1 if the validation loss does not improve over 10 epochs. We use gradient clipping, where the gradients are capped at the maximum gradient norm value of 1. The pretraining converged around 300 epochs at which point the IoU and bIoU on the test split were 0.933 and 0.858 respectively. The pretrained features are evaluated for performance and generalizability on test datasets T1 and T2, where T1 tests performance under varying few-shot schemes and T2 captures the generalizability of the proposed approach under domain shifts. For evaluation of the target task of building segmentation, the pretrained U-Net is modified from the terrain reconstruction task to perform segmentation by adding a segmentation head (convolution block followed by ReLU activation). For finetuning, we use RMS prop optimizer (momentum of 0.999 and weight decay 1×10^{-8}), initial learning rate 1×10^{-6} which decays by a factor of 10 every 15 epochs if the loss does not decrease. Weighted cross entropy loss and DICE loss are used to evaluate segmentation

performance [43]. The model used for pretraining has approximately 24 million parameters and processed approximately 6 images/second on a TWIN-TITAN RTX (2 GPUs). The downstream segmentation model is also of similar complexity with approximately 33 million parameters and processes approximately 10 images/second both during training and inference.

4.3 Segmentation Results

One of the focus areas in this work is label-efficiency, whereby if the pretraining is effective, good performance can be achieved on the downstream task with relatively few labels. We test this by performing segmentation on T1 using nDSM as input instead of DSM. The use of nDSM instead of DSM (as in pretraining) introduces a slight shift in train and test data, making the task a little more challenging. In Table I, we present **ablations** focusing on label percentages, thereby assessing performance under conditions of low-label availability.

Table 1 shows the performance of building segmentation on T1 with different label fractions under full fine-tuning such that all parameters are tuned during this evaluation. Remarkably, with 1% of the labels (equivalent to merely 25 labels), our method achieves similar performance (slightly higher by a margin of 0.03 IoU and similarly for bIoU) compared to pretraining with 1.2 million images and 1,000 different categories. This demonstrates the advantage of learning features from freely available unlabelled data in the domain of remote sensing by curating specific tasks that enable learning features specialized to downstream tasks, to an extent where merely 25 examples are sufficient for target task generalization.

The second and third rows in table 1 show the segmentation performance on 10% (252 examples) and 100% (2500 examples) label fractions. The advantage of pretext pretraining is more apparent in the few-shot schemes (1% and 10%), and as the labels increase, our method closes the gap with ImageNet pretrained features.

	V	Validation			Test			
Approach	IoU	bIoU	Score	IoU	bIoU	Score		
1% (25 labels) on T1 data								
ImageNet	0.742	0.640	0.691	0.772	0.641	0.707		
Ours	0.775	0.551	0.663	0.794	0.507	0.650		
10 % (252 labels) labels on T1 data								
ImageNet	0.851	0.777	0.814	0.848	0.748	0.798		
Ours	0.831	0.731	0.781	0.860	0.769	0.842		
100% (2500 labels) labels on T1 data								
ImageNet	0.901	0.859	0.880	0.9	0.842	0.871		
Ours	0.896	0.834	0.857	0.896	0.835	0.866		

Table 1: Segmentation performance with full fine tuning with different percent of ground truth labels on T1 data compared with our approach against ImageNet pretraining.



Figure 5: Predicted segmentation masks from our approach on T2 (MapAI) dataset.



Figure 6: Images from T2 with labeling inaccuracies and our approach's segmentation masks. As seen in column two, the labels do not accurately reflect the buildings in the images. These inaccuracies are highlighted with orange bounding boxes and signify either absent buildings or buildings marked in the labels but absent in the images. However, despite these errors, our method exhibits robustness in detecting these missing or incorrectly labeled buildings, as observed in the third column. We assume that in the presence of such labels, our reported performance is slightly understated due to incorrect penalization for several of the correct predictions.



Figure 7: Training performance metrics of building segmentation on T2, with network initializations as described in Figure 8.

Generalizability Evaluation

As discussed in Section 1, the other challenge addressed in this work is generalization. In remote sensing, this implies countering shifts that can occur due to different input resolutions, temporal variabilities, labeling variations, as well as errors. In this section, we evaluate the generalizability of pretrained features on T2 [42] which captures a significant distribution shift compared to the training dataset (refer to figure 2). Further, we also test for label efficiency and computational efficiency of our approach. To check for label efficiency we compare with U-Net pretrained on ImageNet classification. To evaluate the computational efficiency and quality of features, we compare against two other approaches that perform building segmentation on T2 [44, 45]. Further, we also perform an ablation against U-Net with random initialization to evaluate the benefit obtained from pretraining on pretext task in terms of downstream convergence speed and ease of optimization (Figures 8 and 7). Our method performs better than ImageNet-based U-Net by a large margin (Table 2; 0.323 in IoU and 0.367 bIoU). Some labels in T2 correspond to real-time labeling errors where buildings in aerial images do not correspond to the available ground truth masks (Figure 6). Further, the masks are generated not from aerial images directly, but from a DTM. As a result, building displacement errors can be expected in the dataset. The low performance of ImageNet pretrained model can be attributed to such label noise in addition to domain shift arising from translating natural image features to LiDAR data. In contrast with [45], which uses both spectral and LiDAR data for building segmentation, we see an improvement by a margin of 0.155 in IoU. This confirms the initial idea that it can be advantageous to use LiDAR data independently, without augmenting it with other spectral information for segmentation problems.

We further show how our pretraining strategy along with a simplistic U-Net allows us to achieve better or comparable performance to significantly more complex architectures [44]. Our terrain-aware U-Net model performs better than the transformer-based segmentation model (Segformer) [46], which leverages a self-attention mechanism to capture long-range dependencies between pixels, which makes it suitable for capturing contextual information in semantic segmentation tasks. This shows that careful pretraining can help achieve computational cost and power efficiency in the long run, allowing much smaller models to achieve similar or better performance. Furthermore, features from the terrainaware pretraining task can be leveraged across a variety of downstream tasks, allowing feature reuse and enhancing label efficiency even further.

Model	IoU	bIoU	Score
U-Net (Random)	0.470	0.261	0.366
U-Net (ImageNet pretrained)	0.521	0.326	0.423
U-Net [44]	0.761	0.582	0.672
U-Net [45]	0.689	0.562	0.625
ConvNext [44]	0.784	0.610	0.697
SegFormer-B0 [44]	0.763	0.590	0.676
SegFormer-B4 [44]	0.784	0.611	0.698
SegFormer-B5 [44]	0.790	0.618	0.704
U-Net (Ours)	0.844	0.693	0.768

 Table 2: Evaluation on Map AI challenge data with full fine tuning

We further compare the three network initializations (random, ImagetNet, and our terrain initializations) to study their impact on training stability and speed. On T2 (Figure 8), our pretraining strategy exhibits greater training stability at the



Figure 8: Training loss of building segmentation on T2, with three different network initializations. *Random Init* refers to initializing the network from random weights, *ImageNet pre-training* refers to initializing the network with weights from training on ImageNet classification. *Ours* refers to initializing the network with weights pretrained on terrain prediction task.

earlier optimization stages. This suggests that our pretraining allows starting from a more optimal point on the loss landscape compared to others. Further, we observe faster convergence, especially when compared to training from random weights. This leads to a more efficient utilization of computational power and resources. The same effect is observed for the performance metrics IoU and bIoU (Figure 7).

5 Conclusion

Motivated by the ever-increasing demand for accurate and upto-date urban planning and development information, in this work we propose leveraging self-supervised learning for efficient building footprint extraction from LiDAR data. Our terrain-aware self-supervised learning task shows improvement with regard to both generalization and label efficiency. Further, we show that pretraining on completely unlabeled data from LiDAR enables the extraction of domain-related features that compete and improve over supervised approaches. In the few-shot evaluation, with as few as 25 labeled examples, our approach rivals or surpasses the traditional ImageNet pretrained models (Tables 1 to 2), bigger architectures (Table 2) and custom-tailored augmentation strategies [45] on building segmentation. Further, as opposed to approaches combining data from different sensors, in this work we use LiDAR data independently, indicating the usefulness of depth information inherent in LiDAR for building segmentation. The data used in this article can be accessed at https://dataverse.no/ (DOI: 10.18710/HSMJLL).

Acknowledgment

This paper was funded by the Research Council of Norway under the project title ATELIER-EO: Automated machine learning framework tailored to Earth Observation [project number : 336990].

References

- X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.
- [2] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark," in 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2017, pp. 3226–3229.
- [3] A. Janik and K. Sankaran, "Sampling strategy for finetuning segmentation models to crisis area under scarcity of data," *arXiv preprint arXiv:2202.04766*, 2022.
- [4] B. Uzkent, E. Sheehan, C. Meng, Z. Tang, M. Burke, D. Lobell, and S. Ermon, "Learning to interpret satellite images using wikipedia," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.
- [5] G. Chu, B. Potetz, W. Wang, A. Howard, Y. Song, F. Brucher, T. Leung, and H. Adam, "Geo-aware networks for fine-grained recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.
- [6] K. Ayush, B. Uzkent, C. Meng, K. Tanmay, M. Burke, D. Lobell, and S. Ermon, "Geography-aware selfsupervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10181–10190.
- [7] H. Dong, W. Ma, Y. Wu, J. Zhang, and L. Jiao, "Selfsupervised representation learning for remote sensing image change detection based on temporal prediction," *Remote Sensing*, vol. 12, no. 11, p. 1868, 2020.
- [8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer, 2015, pp. 234–241.
- [9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [10] A. A. Aleissaee, A. Kumar, R. M. Anwer, S. Khan, H. Cholakkal, G.-S. Xia, and F. S. Khan, "Transformers in remote sensing: A survey," *Remote Sensing*, vol. 15, no. 7, p. 1860, 2023.
- [11] C. Hug, P. Krzystek, and W. Fuchs, "Advanced lidar data processing with lastools," in *IAPRS*, vol. XXXV. ISPRS, 7 2004. [Online]. Available: https://www.isprs.org/ proceedings/XXXV/congress/comm2/papers/240.pdf

- [12] Z. Li, M. E. Hodgson, and W. Li, "A generalpurpose framework for parallel processing of largescale lidar data," *International Journal of Digital Earth*, vol. 11, pp. 26–47, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:46764144
- [13] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios, "Building detection in very high resolution multispectral data with deep learning features," in 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2015, pp. 1873–1876.
- [14] G. Prathap and I. Afanasyev, "Deep learning approach for building detection in satellite multispectral imagery," in 2018 International Conference on Intelligent Systems (IS), 2018, pp. 461–465.
- [15] L. Ivanovsky, V. Khryashchev, V. Pavlov, and A. Ostrovskaya, "Building detection on aerial images using u-net neural networks," in 2019 24th Conference of Open Innovations Association (FRUCT), 2019, pp. 116–122.
- [16] H. Miyazaki, K. Kuwata, W. Ohira, Z. Guo, X. Shao, Y. Xu, and R. Shibasaki, "Development of an automated system for building detection from high-resolution satellite images," in 2016 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA), 2016, pp. 245–249.
- [17] M. Aamir, Y.-F. Pu, Z. Rahman, M. Tahir, H. Naeem, and Q. Dai, "A framework for automatic building detection from low-contrast satellite images," *Symmetry*, vol. 11, no. 1, 2019. [Online]. Available: https: //www.mdpi.com/2073-8994/11/1/3
- [18] G. Sohn and I. Dowman, "Data fusion of high-resolution satellite imagery and lidar data for automatic building extraction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, no. 1, pp. 43–63, 2007.
- [19] T. Hermosilla, L. A. Ruiz, J. A. Recio, and J. Estornell, "Evaluation of automatic building detection approaches combining high resolution images and lidar data," *Remote Sensing*, vol. 3, no. 6, pp. 1188–1210, 2011.
- [20] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 574– 586, 2018.
- [21] D. Yu, S. Ji, J. Liu, and S. Wei, "Automatic 3d building reconstruction from multi-view aerial images with deep learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 171, pp. 155–170, 2021.
- [22] M. Alonso and J. Malpica, "Satellite imagery classification with lidar data," *Trees (A)*, vol. 2106, no. 19, p. 11, 2010.
- [23] P. Goyal, M. Caron, B. Lefaudeux, M. Xu, P. Wang, V. Pai, M. Singh, V. Liptchinsky, I. Misra, A. Joulin *et al.*, "Self-supervised pretraining of visual features in the wild," *arXiv preprint arXiv:2103.01988*, 2021.

- [24] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference* on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.
- [25] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv* preprint arXiv:2003.04297, 2020.
- [26] S. Zhang, Z. Wen, Z. Liu, and Q. Pan, "Rotation awareness based self-supervised learning for sar target recognition," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 1378–1381.
- [27] Z. Zhao, Z. Luo, J. Li, C. Chen, and Y. Piao, "When selfsupervised learning meets scene classification: Remote sensing scene classification based on a multitask learning framework," *Remote Sensing*, vol. 12, no. 20, p. 3276, 2020.
- [28] C. Tao, J. Qi, W. Lu, H. Wang, and H. Li, "Remote sensing image scene classification with self-supervised paradigm under limited labeled samples," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.
- [29] S. Vincenzi, A. Porrello, P. Buzzega, M. Cipriano, P. Fronte, R. Cuccu, C. Ippoliti, A. Conte, and S. Calderara, "The color out of space: learning self-supervised representations for earth observation imagery," in 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, pp. 3034–3041.
- [30] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
- [31] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remotesensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645– 657, 2016.
- [32] J. Yuan, "Learning building extraction in aerial scenes with convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 11, pp. 2793–2798, 2017.
- [33] G. Wu, X. Shao, Z. Guo, Q. Chen, W. Yuan, X. Shi, Y. Xu, and R. Shibasaki, "Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks," *Remote Sensing*, vol. 10, no. 3, p. 407, 2018.
- [34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

- [35] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.
- [36] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE* International Conference on Computer Vision, 2015, pp. 1440–1448.
- [37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
- [38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [40] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. IEEE, 2003, pp. 1398–1402.
- [41] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
- [42] S. Jyhne, M. Goodwin, P. Andersen, I. Oveland, A. Nossum, K. Ormseth, M. Ørstavik, and A. Flatman, "MapAI: Precision in building segmentation. Nordic Machine Intelligence 2022 sep; 2 (3): 1–3. doi: 10.5617/nmi. 9849."
- [43] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis* and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3. Springer, 2017, pp. 240– 248.
- [44] L. Li, T. Zhang, S. Oehmcke, F. Gieseke, and C. Igel, "Buildseg: A general framework for the segmentation of buildings," *arXiv preprint arXiv:2301.06190*, 2023.
- [45] K. A. Borgersen and M. Grundetjern, "MapAI competition submission for team Kaborg: Using stable diffusion for ml image augmentation," *Nordic Machine Intelligence*, vol. 2, no. 3, 2022.
- [46] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.