Semi-Supervised Learning in Aerial Imagery: Implementing Uni-Match with Frame Field learning for Building Extraction

Patrik Sabol[1], Bibiána Lajčinová[2]

[1]*Geodeticca Vision s.r.o., Floriánska 19, 044 01 Košice, Slovak republic*

[2]*Slovak National Supercomputing Center, Dúbravská cesta 3484/9, 84104 Bratislava-Karlova Ves, Slovak republic*

Introduction

Building extraction in GIS (geographic information system) is pivotal for urban planning, environmental studies, and infrastructure management, allowing for accurate mapping of structures, including the detection of illegal constructions for regulatory compliance. Integrating extracted building data with other geospatial layers enhances the understanding of urban dynamics and spatial relationships. Given the scale and complexity of these tasks, there is a growing need to automate building extraction using deep learning techniques, which offer improved accuracy and efficiency in handling large-scale geospatial data.

State-of-the-art image segmentation models primarily output in raster format, whereas GIS applications often require vector polygons. One such method to meet this requirement is Frame Field learning, which addresses the gap between raster format outputs of image segmentation models and the vector format needed in GIS. This approach significantly enhances the accuracy of building vectorization by aligning with ground truth contours and provide topologically clean vector objects.

These models are trained using a 'supervised learning' method, necessitating a large amount of labeled examples for training. However, obtaining such a significant volume of data can be extremely challenging and expensive. A potential solution to this problem is 'semi-supervised learning,' a method that reduces reliance on labeled data. In semi-supervised learning, the model is trained with a mix of a small set of labeled data and a larger set of unlabeled data. Hence, the goal of this collaboration between the Slovak National Competence Center for High-Performance Computing and Geodeticca Vision s.r.o. was to identify, implement, and evaluate an appropriate semi-supervised method for Frame Field learning.

Methods

Frame Field learning

The key idea of the frame field learning [1] is to help the polygonization method in solving ambiguous cases caused by discrete probability maps (output from image segmentation models). This is accomplished by introducing an additional output to the neural network of image segmentation, namely a frame field (see. Fig. 1), which represents the structural features and geometrical characteristics of the building.

Frame fields

Frame field is a 4-PolyVector field that assigns four vectors to each point on a plane. Specifically, the first two vectors are constrained to be opposite to the other two, meaning each point is assigned a set of vectors {u, −u, v, −v}. This approach is particularly necessary for buildings, as they are regular structures with sharp corners, and capturing directionality at these sharp corners requires two directions.

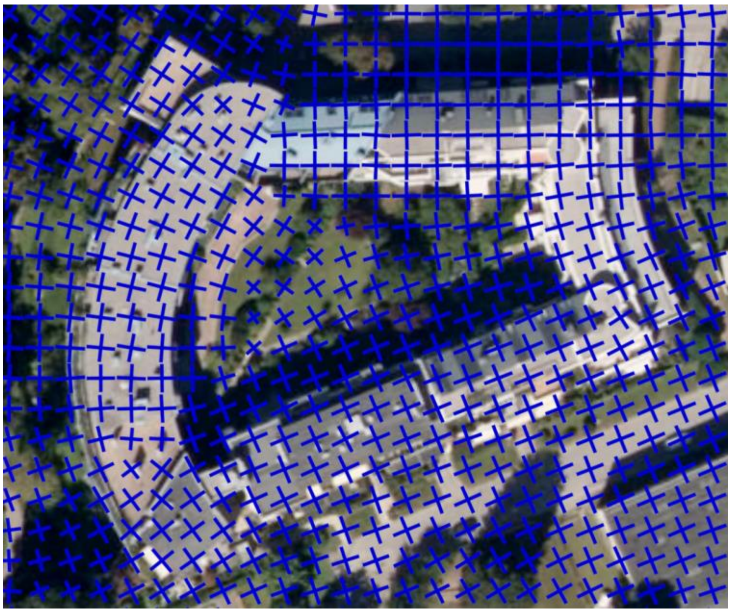


Figure 1: Visualization of the frame field output on the image from training set [1].

Frame Field learning

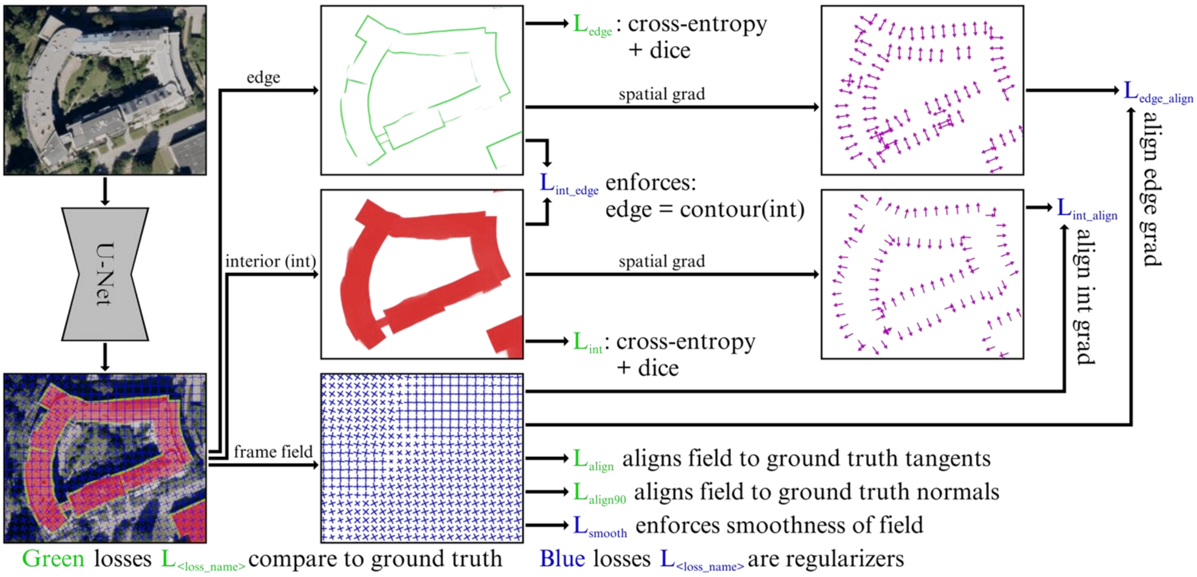

edge
U-Net
interior (int)
frame field
L_{edge}: cross-entropy + dice
spatial grad
L_{int_edge} enforces: edge = contour(int)
spatial grad
L_{int}: cross-entropy + dice
L_{edge_align}
align edge grad
L_{int_align}
align int grad
L_{align} aligns field to ground truth tangents
$L_{align90}$ aligns field to ground truth normals
L_{smooth} enforces smoothness of field
Green losses $L_{<loss\_name>}$ compare to ground truth
Blue losses $L_{<loss\_name>}$ are regularizers

Figure 2: Diagram of the frame field learning [1]

The learning process of frame fields can be summarized as follows:

1. The network's input is a 3×H×W RGB image.
2. To generate a feature map, any deep segmentation model could be used, such as U-Net, which is then processed to output detailed segmentation maps.
3. The training is supervised with ground truth rasterized polygons for interiors and edges, utilizing a mix of cross-entropy and Dice loss for accurate segmentation.

4. To train the frame field, three losses are used:
    1. L_{align} enforces alignment of the frame field to the tangent direction.
    2. $L_{align90}$ prevents the frame field from collapsing to a line field.
    3. L_{smooth} measures the smoothness of the frame field.
5. Additional losses, regularization losses, are introduced to maintain output consistency, aligning the spatial gradients of the predicted maps with the frame field.

Vectorization

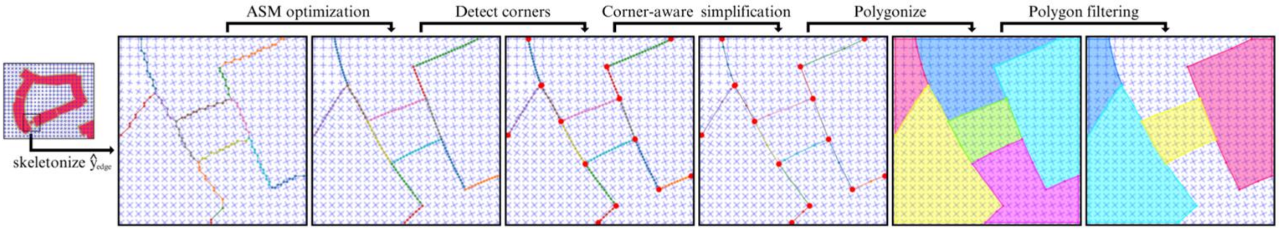


Figure 3: Visualization of the vectorization process [1]

The vectorization process transforms classified raster images into vector polygons using a polygonization method using the Active Skeleton Model (ASM). The principle of this algorithm is the iterative shifting of the vertices of the skeleton graph to their ideal positions. This method optimizes a skeleton graph - a network of pixels outlining the building's structure - created by a thinning method applied on a building wall probability map. The iterative shifting is controlled by a gradient optimization method aimed at minimizing an energy function, which includes specific components related to the structure and geometry being analyzed:

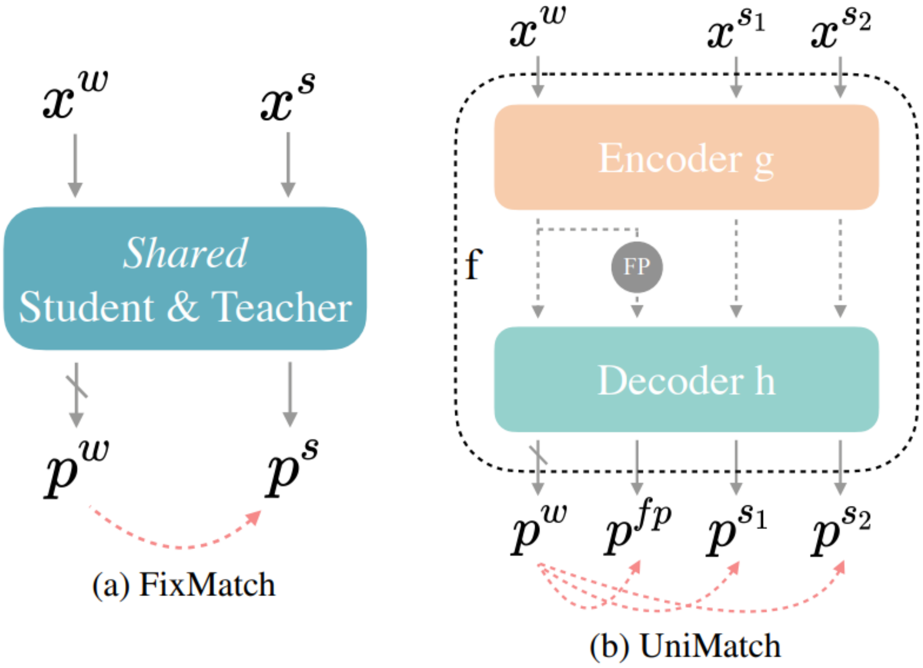
1. $E_{probability}$ – fits the skeleton paths to the contour of the building interior probability map at a certain probability threshold, e.g. 0.5
2. $E_{frame\ field\ align}$ - aligns each edge of the skeleton graph to the frame field.
3. E_{length} – ensures that the node distribution along paths remains homogeneous as well as tight.

UniMatch semi-supervised learning

UniMatch [2], an advanced semi-supervised learning method in the consistency regularization category, builds upon the foundational principles established by FixMatch [3], a baseline method in this domain. primarily operates on the principle of pseudo-labeling combined with consistency regularization.

The basic principle of the FixMatch method involves generating pseudo-labels for unlabeled data from the predictions of a neural network. Specifically, for a weakly perturbed unlabeled input x^w, a prediction p^w is generated, which serves as a pseudo-label for the prediction of x^s, a strongly perturbed input. Subsequently, the loss function value, for example, cross-entropy($p^{w,}$ p^s), is calculated, considering only areas from p^w with a probability value greater than a certain threshold, e.g., >0.95. UniMatch builds upon and extends the FixMatch methodology, introducing two core enhancements:

1. UniPerb (Unified Perturbations for Images and Features) - This involves applying perturbations at the feature level. Practically, this means applying a dropout function to the output (i.e., the feature) from the encoder layer of the neural network, randomly ignore features, which then proceed to the decoder part of the network, generating p^{fp}.



(a) FixMatch

(b) UniMatch

Figure 4: (a) The FixMatch baseline (b) used UniMatch method. The FP denotes feature pertubation, w and s means weak and strong pertubation, respectively [2].

2. DusPerb (Dual-Stream Perturbations) – Instead of using one strong perturbation, two perturbations are utilized.

Ultimately, there are three error functions: crossentropy($p^{w,}$ p^{fp}), crossentropy($p^{w,}$ p^{s1}), crossentropy($p^{w,}$ p^{s2}). These are then linearly combined with the supervised error function.

Integrating UniMatch Semi-Supervised Learning with Frame Field Learning

Implementation Strategy for UniMatch in Frame Field Learning

To integrate UniMatch into our Frame Field learning framework, we first differentiated between weak and strong perturbations. For weak perturbations, we chose basic spatial transformations such as rotation, mirroring, and vertical/horizontal flips. These are well-suited for aerial imagery and straightforward to implement.

For strong perturbations, we opted for photometric transformations. These include adjustments in hue, color, and brightness, providing a more significant alteration to the images compared to spatial transformations.

Incorporating feature perturbation loss was a crucial step. We implemented this by introducing a dropout mechanism between the encoder and decoder parts of the network. This dropout selectively omits features at the feature level, which is essential for the UniMatch approach.

Regarding the dual-stream perturbations of UniMatch, we adapted our model to handle two types of strong perturbations. The dual-stream approach involves using the weak perturbation prediction as a pseudo-label and training the model using the strong perturbation predictions as loss functions. We have two strong perturbations, hence the term 'dual-stream'. Each of these perturbations contributes to the overall robustness and effectiveness of the model in semi-supervised learning scenarios, especially in the context of building extraction from complex aerial imagery. Through these specific implementations, UniMatch was successfully integrated into the Frame Field learning model, enhancing its capability to efficiently process and learn from both labeled and unlabeled data.

Experiments

Dataset

Labeled Data

Our labeled data comes from three different sources, which we'll detail in the accompanying Table 1.

Name	Dataset Type	Area Covered (sq km)	Number of Buildings	Orthophotography Resolution (m)	Tile Resolution (px)
Geodeticca-Buildings	Private	198.18	50,354 buildings	0.25	12500x10000
INRIA [4]	Open source	810.00	206,679 buildings	0.30	5000x5000
Landcover.ai [5]	Open source	216.27	12,354 buildings	0.25/0.50	9000x9500 / 4200x4700

Table 1: Overview of 3 data sources of labeled data used for training the models with details.

Unlabeled Data

For the unlabeled dataset, we selected high-quality aerial images from Geodetický a kartografický ústav (GKÚ) [6], available for free public use. We specifically targeted a diverse area of 7000 square kilometers, ensuring a wide representation of various landscapes and urban settings.

Data Processing: Patching

We processed both labeled and unlabeled images into patches of size 320x320 px. This patch size is specifically chosen to match the input requirements of our neural network. From the labeled data, this process resulted in approximately 55,000 patches. Similarly, from the unlabeled dataset, we obtained around 244,000 patches.

Training setup

Model Architecture

We designed our model using a U-Net architecture with an EfficientNet-B4 backbone. This combination provides a good balance of accuracy and efficiency, crucial for handling the complexity of our segmentation tasks. The EfficientNet-B4 backbone was specifically chosen for its optimal balance between memory usage and performance. In Frame Field learning, U-Net architecture has been shown to be highly effective, as evidenced by its strong performance in prior studies.

Training Process

For training, we used the AdamW optimizer, which combines the advantages of Adam optimization with weight decay, aiding in better model generalization. To prevent overfitting, we implemented L2 regularization. Additionally, we used the ReduceLROnPlateau learning rate scheduler. This scheduler adjusts the learning rate based on validation loss, ensuring efficient training progress.

Semi-Supervised Learning Adjustments

A key aspect of our training was adjusting the ratio of unlabeled to labeled patches. We experimented with ratios ranging from 1:1 to 1:5 (labeled:unlabeled). This variability allowed us to explore the impact of different amounts of unlabeled data on the learning process. It enabled us to identify the optimal balance for training our model, ensuring effective learning while leveraging the advantages of semi-supervised learning in handling large and diverse datasets.

Model evaluation

In our evaluation of the building footprint extraction model, we chose metrics that precisely measure how well our predictions align with real-world structures.

Intersection over Union (IoU)

Intersection over Union (IoU) is a key metric we used. It calculates the overlap between our model's predictions and the actual building shapes. An IoU score close to 1 means our predictions closely match the real buildings. This metric is essential for assessing the geometric accuracy of the segmented areas, directly reflecting the precision of boundary delineation. Moreover, by evaluating the ratio of correctly predicted area to the combined area of prediction and ground truth, IoU provides a clear measure of the model's effectiveness in capturing the true extent and shape of buildings in complex urban landscapes.

Precision, Recall and F1

Precision measures the accuracy of the model's building predictions, indicating the proportion of correctly identified buildings out of all identified buildings, thereby reflecting the model's specificity. Recall assesses the model's ability to capture all actual buildings, with a high recall score highlighting its sensitivity in detecting buildings. The F1 Score combines precision and recall into a single metric, offering a balanced view of the model's performance by ensuring that high scores result from both high precision and high recall.

Complexity Aware IoU (cIoU)

We also utilized Complexity Aware IoU (cIoU) [7]. This metric addresses a shortfall in IoU by balancing segmentation accuracy and the complexity of the polygon shapes. While IoU alone can lead models to create overly complex polygons, cIoU ensures that the complexity of the polygons (number of vertices) is kept realistic, reflecting the typically less complex structure of real buildings.

N Ratio Metric

The N ratio metric was an additional component of our evaluation strategy. It contrasts the number of vertices in our predicted shapes with those in the actual buildings [7]. This helps in understanding whether our model accurately replicates the detailed structure of the buildings.

Max Tangent Angle Error

To ensure clean geometry in building extraction tasks, accurately measuring contour regularity is essential. The Max Tangent Angle Error (MTAE) [1] metric is designed to address this need by supplementing the Intersection over Union (IoU) metric. It specifically targets the limitation of IoU, where segmentations with rounded corners may receive higher scores than those with more precise, sharp corners. By evaluating the alignment of edges through the comparison of tangent angles at sampled points along predicted and ground truth contours, MTAE effectively penalizes inaccuracies in edge orientation. This focus on edge precision is critical for producing clean vector representations of buildings, emphasizing the importance of accurate edge delineation in segmentation tasks.

Evaluation Process

Trained models were tested on large, full-size aerial images, instead of small patches. This method ensured that our evaluation closely mirrored real-world scenarios. To extract buildings from full-size images, we employed a sliding window technique, aligning with the neural network's training on specific patch sizes, to systematically produce segment-wise predictions. An advanced averaging technique was applied at the borders of these overlapping segments, crucial for minimizing artifacts and ensuring consistency across the prediction map. The resulting seamless, full-size output was then vectorized into precise vector polygons using the Active Skeleton Model (ASM) algorithm.

Results

Training Approach	Ratio (Labeled:Unlabeled)	IoU (%)	Precision (%)	Recall (%)	F1 Score (%)	N Ratio	cIoU (%)	Mean MTAE(°)
Baseline	-	80.50	85.75	94.27	89.81	2.33	48.89	18.60
Semi-Supervised	1:1	83.88	87.66	93.41	90.44	1.94	56.98	20.47
Semi-Supervised	1:3	85.35	90.04	94.25	92.10	1.76	61.91	18.92
Semi-Supervised	1:5	85.77	90.04	94.76	92.34	1.65	64.75	17.45

Table 2: Results of the models' training for baseline (supervised) a semi-supervised approaches with different ratios of labeled to unlabeled images used.

The results from our experiments, reflecting performance of segmentation model trained under different conditions, reveal significant insights (see Table 2). We evaluated the model's performance in a baseline scenario without semi-supervised learning and in scenarios where semi-supervised learning was applied with varying ratios of labeled to unlabeled data (1:1, 1:3, and 1:5).

1. **IoU Percentage Increase:** Starting from the baseline IoU of 80.50%, we observed a steady increase in this metric as we introduced more unlabeled data into the training process, reaching up to 85.77% with a 1:5 labeled to unlabeled ratio.
2. **Precision, Recall, and F1 Score:** The precision of the model, which measures how accurate the predictions are, improved from 85.75% in the baseline to 90.04% in the 1:5 ratio setup. Similarly, recall, which indicates how well the model can find all relevant instances, slightly increased from 94.27% to 94.76%. The F1 Score, which balances precision and recall, also

saw an improvement from 89.81% to 92.34%. These improvements suggest that the model became more accurate and reliable in its predictions when semi-supervised learning was used.

3. **N Ratio and cIoU:** The results show a notable decrease in the N Ratio from 2.33 in the baseline to 1.65 in the semi-supervised 1:5 ratio setup, indicating that the semi-supervised model generates simpler, yet accurate, vector shapes that more closely resemble the actual structures. This simplification likely contributes to the enhanced usability of the output in practical GIS applications. Concurrently, the complexity-aware IoU (cIoU) significantly improved from 48.89% in the baseline to 64.75% in the 1:5 ratio, suggesting that the semi-supervised learning approach not only improves the overlap between the predicted and actual building footprints but also produces simpler vector shapes, which are closer to real-world buildings in terms of geometry.

4. **Mean Max Tangent Angle Error (MTAE):** The Mean MTAE's reduction from 18.60° in the baseline to 17.45° in the 1:5 semi-supervised setting signifies an improvement in the geometric precision of the model's predictions. This suggests that the semi-supervised learning model is better at capturing the architectural features of buildings with more accurately defined angles, contributing to the production of topologically simpler and cleaner vector polygons.

Training on High-Performance Computing (HPC) Machine

HPC Configuration

Our training was conducted on a High-Performance Computing (HPC) machine equipped with substantial computational resources. The HPC had 8 nodes, each outfitted with 4 NVIDIA A100 GPUs with 40GB of VRAM, 64 CPU cores, and 256GB of RAM. For task scheduling, the system utilized Slurm.

PyTorch Lightning Framework

We employed the PyTorch Lightning framework, which offers user-friendly multi-GPU settings. This framework allows the specification of the number of GPUs per node, the total number of nodes, various distributed strategies, and the option for mixed-precision training.

Experiences with Slurm and PyTorch Lightning

When training on a single GPU, our Slurm configuration was as follows:

```
#SBATCH --partition=ngpu
#SBATCH --gres=gpu:1
#SBATCH --cpus-per-task=16
#SBATCH –mem=64000
```

In PyTorch Lightning, we set the trainer as:

```
trainer = Trainer(accelerator="gpu", devices=1)
```

Since, here, we allocated one GPU from four available in one node, we allocated 16 CPUs from 64 available. Therefore, for the data loaders, we assigned 16 workers. Since semi-supervised learning uses two data loaders (one for labeled and one for unlabeled data), we allocated 8 workers to each. It was critical to ensure that the total number of cores for the data loaders did not exceed the available CPUs to prevent training crashes.

Distributed Data Parallel (DDP) Strategy

Using PyTorch Lightning's Distributed Data Parallel (DDP) option, we ensured each GPU across the nodes operated independently:

- Each GPU processed a portion of the dataset.
- All processes initiated the model independently.
- Each conducted forward and backward passes in parallel.
- Gradients were synchronized and averaged across processes.
- Each process updated its optimizer individually.

With this approach, the total number of data loaders equaled the number of GPUs multiplied by the number of data loaders. For example, in a semi-supervised learning setup with 4 GPUs and two types of data loaders (labeled and unlabeled), we ended up with 8 data loaders, each with 8 workers – 64 workers in total.

To fully utilized one node with four GPU, we used following configurations:

```
#SBATCH --partition=ngpu
#SBATCH --gres=gpu:4
#SBATCH –exclusive – it means to use all resources of the node
#SBATCH --cpus-per-task=64 – no need to set since it uses all resources
#SBATCH –mem=256000 – no need to set since it uses all resources
```

In PyTorch Lightning, we set the trainer as:

```
trainer = Trainer(accelerator="gpu", devices=4, strategy="ddp")
```

Utilizing Multiple Nodes

Using PyTorch Lighting, it is possible to leverage multiple nodes on HPC. For instance, using 4 nodes with 4 GPUs each (16 GPUs in total) was configured as:

```
trainer = Trainer(accelerator="gpu", devices=4, strategy="ddp", num_nodes=4)
```

Correspondingly, the Slurm configuration was set to:

```
#SBATCH –nodes=4
#SBATCH –ntasks-per-node=4
#SBATCH --gres=gpu:4
```

These settings and experiences highlight the scalability and flexibility of training complex machine learning models on an HPC environment, especially for tasks demanding significant computational resources like semi-supervised learning in geospatial data analysis.

Training Scalability Analysis

Training type	# GPU	Time per epoch (min)	Speedup Ratio (vs. 1 GPU)

Supervised learning	8 (2 nodes)	1:25	5.56x
	4	2:38	2.99x
	2	4:01	1.96x
	1	7:53	1.00x
Semi-supervised learning with ratio 1:1	8 (2 nodes)	3:55	7.25x
	4	7:17	3.90x
	2	14:24	1.97x
	1	28:23	1.00x

Table 3: Training results of supervised and semi-supervised approaches with 1, 2, and 4 GPUs. Time for one epoch is reported for each configuration.

In the Training Scalability Analysis, we carefully examined the impact of expanding computational resources on the efficiency of training models, utilizing the PyTorch Lightning framework. This investigation covered both supervised and semi-supervised learning approaches, with a particular emphasis on the effects of increasing GPU numbers, including setups involving 2 nodes (or 8 GPUs).

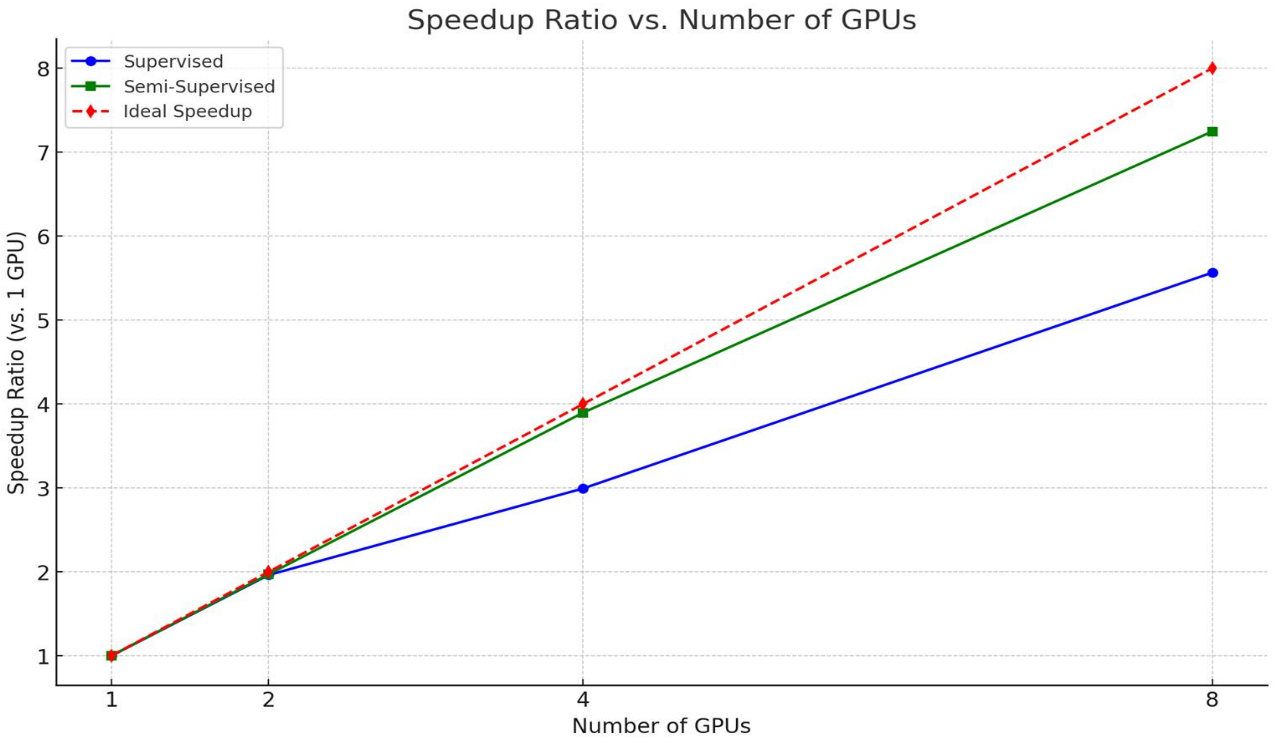


Figure 5: This graph compares the actual speedup ratios for supervised and semi-supervised learning against the number of GPUs, alongside the ideal linear speedup ratio. It showcases the closer alignment of semi-supervised learning with ideal scalability, emphasizing its greater efficiency gains from increased computational resources.

A key finding from this analysis was that the increase in speedup ratios for supervised learning did not perfectly align with the number of GPUs utilized. Ideally, doubling the number of GPUs would directly double the speedup ratio (e.g., using 4 GPUs would result in a 4x speedup). However, the actual speedup ratios were lower than this ideal expectation. This discrepancy can be attributed to the overhead associated with managing multiple GPUs and nodes, particularly the need to synchronize data across all GPUs, which introduces efficiency losses.

In contrast, semi-supervised learning exhibited a trend that more closely approached the ideal linear increase in speedup ratios. The complex nature and higher computational requirements of semi-supervised learning tasks seem to diminish the relative effect of overhead costs, thereby allowing for a more efficient use of additional GPUs. Despite the challenges of data synchronization across multiple GPUs and nodes, the intensive computational demands of semi-supervised learning enable a more effective scaling of resources, yielding speedup ratios that more closely mirror the ideal scenario.

Conclusion

The research presented in this whitepaper has successfully demonstrated the effectiveness of integrating UniMatch semi-supervised learning with Frame Field learning for the task of building extraction from aerial imagery. This integration addresses the challenges associated with the scarcity of labeled data in deep learning applications for geographic information systems (GIS), providing a cost-effective and scalable solution.

Our findings reveal that employing semi-supervised learning significantly enhances the model's performance across several key metrics, including Intersection over Union (IoU), precision, recall, F1 Score, N Ratio, complexity-aware IoU (cIoU), and Mean Max Tangent Angle Error (MTAE). Notably, the improvements in IoU and cIoU metrics underscore the model's increased accuracy in delineating building footprints and generating vector shapes that closely resemble actual structures. This outcome is pivotal for applications in urban planning, environmental studies, and infrastructure management, where precise mapping and analysis of building data are crucial.

The methodology adopted, which combines Frame Field learning with the innovative UniMatch approach, has proven to be highly effective in leveraging both labeled and unlabeled data. This strategy not only improves the geometric precision of the model's predictions but also ensures the generation of cleaner, topologically accurate vector polygons. Furthermore, the scalability and efficiency of training on a High-Performance Computing (HPC) machine using the PyTorch Lightning framework and Distributed Data Parallel (DDP) strategy have been instrumental in handling the extensive computational demands of the semi-supervised learning process on the data at hand, within a time frame ranging from tens of minutes to hours.

In conclusion, this research highlights the potential of semi-supervised learning in significantly advancing the field of automated building extraction from aerial imagery. The implementation of UniMatch within the Frame Field learning framework represents a notable leap forward, offering a robust solution to the challenges of data scarcity and the need for high-precision geospatial data analysis. This approach not only enhances the efficiency and accuracy of building extraction but also opens new avenues for the application of semi-supervised learning techniques in GIS and related fields.

Acknowledgements

Research results were obtained with the support of the Slovak National competence centre for HPC, the EuroCC 2 project and Slovak National Supercomputing Centre under grant agreement 101101903-EuroCC 2-DIGITAL-EUROHPC-JU-2022-NCC-01. Computational resources were procured in the national project National competence centre for high performance computing (project code: 311070AKF2) funded by European Regional Development Fund, EU Structural Funds Informatization of society, Operational Program Integrated Infrastructure.

References:

[1] Nicolas Girard, Dmitriy Smirnov, Justin Solomon, and Yuliya Tarabalka. "Polygonal Building Extraction by Frame Field Learning". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2021), pp. 5891-5900.

[2] L. Yang, L. Qi, L. Feng, W. Zhang, and Y. Shi. "Revisiting Weak-to-Strong Consistency in Semi-Supervised Semantic Segmentation". In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2023), pp. 7236-7246. doi: 10.1109/CVPR52729.2023.00699.

[3] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. "FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence". In: CoRR, vol. abs/2001.07685 (2020). Available: https://arxiv.org/abs/2001.07685.

[4] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. "Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark". In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (2017). IEEE.

[5] Adrian Boguszewski, Dominik Batorski, Natalia Ziemba-Jankowska, Tomasz Dziedzic, and Anna Zambrzycka. "LandCover.ai: Dataset for Automatic Mapping of Buildings, Woodlands, Water and Roads from Aerial Imagery". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2021), pp. 1102-1110.

[6] "Ortofotomozaika." Geoportal SK. Accessed February 14, 2024. https://www.geoportal.sk/sk/zbgis/ortofotomozaika/.

[7] Stefano Zorzi, Shabab Bazrafkan, Stefan Habenschuss, and Friedrich Fraundorfer. "PolyWorld: Polygonal Building Extraction with Graph Neural Networks in Satellite Images". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022), pp. 1848-1857.