GeoAI for Topographic Data Accuracy Enhancement: the AI4TDB project

Lingli Zhu, Jere Raninen, Emilia Hattula

National Land Survey of Finland (NLS) - lingli.zhu@nls.fi, jere.raninen@nls.fi, emilia.hattula@nls.fi

Keywords: GeoAI, building detection, topographic database, deep learning, neural network, building footprint

Abstract

GeoAI combines artificial intelligence (AI) with geospatial data, science, and technologies. In this paper, a successful case in utilizing GeoAI to improve the data quality in the national topographic database (TDB) of Finland was introduced. The project employed a GeoAI model to identify buildings from the input data of true orthophotos, digital elevation model (DTM), and digital surface model (DSM). The GeoAI-derived buildings served as reference data, enabling a comparison with building polygons from the topographic database (TDB) to reveal TDB building location deviations, missing structures, and demolished buildings. Throughout the project, algorithms were developed to match the TDB building vectors to the GeoAI-derived building polygons. The challenges include i) the differences between the GeoAI-derived building outlines and the TDB build footprints; ii) the reliability of the GeoAI model across data over different environments: urban, suburban, rural, and forest areas. Throughout the project, the GeoAI model was continuously improved by training massive new datasets: corrected vectors from the model prediction. Testing was conducted on datasets from twelve areas of Finland, covering 2204 km2 and including hundreds of thousands of buildings. The test areas covered urban, suburban, rural, and forest areas. The evaluation was conducted by the mapping team in the organization. The results showed that our methods greatly enhanced the quality of the TDB building footprints. The challenges and lessons of the project were addressed in the paper.

1. Introduction

In recent years, the GeoAI application has been paid high attention. The availability of high-resolution geographic data and high-performance computing techniques together with deep learning fuel progress in fast and accurate object detection (Janowicz et al., 2020). VoPham et al. (2018) gave an overview of GeoAI technologies for exposure modeling in environmental epidemiology, including the capability to incorporate large amounts of spatial big data of high spatial and/or temporal resolution; computational efficiency regarding time and resources; flexibility in accommodating important features of spatial (environmental) processes. Liu & Biljecki (2022) reviewed spatially-explicit GeoAI in Urban Geography including Urban Dynamics, Social Differentiation of Urban Areas, and Social Sensing. The authors revealed that although GeoAI was a trending topic in geography and the applications of deep neural network-based methods were proliferating, the development of spatially-explicit GeoAI models was still in their early phase. This paper will address a new application: GeoAI for improving topographic building data accuracy.

The growth of the urban population has accelerated building and other infrastructure construction as well as the number of built-up areas. Although the information in the national topographic database (TDB) updates yearly, owing to historical factors, the vectors may contain inaccuracies. For instance, considering the building vectors—decades ago, analog photogrammetric instruments were used to measure buildings from film-derived stereo images. The evolution to digital photogrammetry workstations and subsequent advancements in digital cameras drastically improved aerial image resolution, accompanied by robust software and hardware facilitating precise calculations. Yet, revisiting earlier measurements across different periods proved impractical. Consequently, early-stage building footprints might harbor a degree of error. The errors of TDB building footprints have been evidenced in some projects in the National Land Survey of Finland (NLS) when they overlapped with Lidar point clouds. The errors included deviations in building locations, instances of missed buildings, and cases of demolished structures.

Furthermore, the national topographic database has served the public as open data for over a decade. Its data quality and accuracy have considerably influenced on research and development of academies and industry sectors. Under such contexts, the AI4TDB project was launched. ‘AI4TDB’ was the abbreviation of 'Artificial Intelligence for Topographic Database Accuracy Enhancement'. The project aimed to leverage artificial intelligence to improve the precision of the Topographic Database and ensure the up-to-dateness of spatial data within the NLS TDB. The project focused on two types of objects: buildings and watercourses. In this paper, we focus on the buildings.

Prior to the AI4TDB project, we developed a deep learning solution, utilizing UNet++ for precise building detection from the training datasets of true orthophotos, building vectors, LidarDTM, and LidarDSM, yielding good accuracy levels evaluated by the NLS expert. The AI4TDB project capitalized on these advancements by using GeoAI-derived buildings as references to rectify errors in TDB building footprints. Additionally, the GeoAI model for building detection, initially trained by the former project, underwent further refinement during the AI4TDB project. This involved training the model with 2023 datasets sourced from multiple production areas, resulting in notable enhancements in the performance of the GeoAI model.

2. Literature Review

The AI4TDB project employed a UNet++ model, which has been developed and trained for building detection from previous project. Although the method will be reported with details in a separate paper, an overview of the state of the art in building detection will be introduced in the following paragraphs regarding the use of datasets and the development of methods.

The datasets for building detection mainly include imagery, point cloud, and their fusion. Building detection from imagery included using satellite imagery (Sirko et al., 2021), aerial imagery (Saito & Aoki, 2015), orthophotos (Dornaika et al., 2016), true orthophotos (Buyukdemircioglu et al., 2022), UAV imagery (Ham et al., 2018; Boonpook et al., 2018), and terrestrial imagery (Chaloeivoot & Phiphobmongkol, 2016).

Building detection from point clouds included using airborne Lidar (Sun & Salvaggio, 2013), terrestrial laser scanning (Jiang et al., 2018), mobile laser scanning (Li et al., 2016), or photogrammetric point cloud (Malihi et al., 2016). Examples of building detection using the fusion of imagery and point cloud can be found in Gilani et al. (2016), Nahhas et al.(2018), and Ghasemian et al. (2022). They all employed Lidar data and orthophotos. Lidar data and imagery possess complementary characteristics. While imagery typically includes shadows and is presented in a 2D raster format, it indicates smooth building edges. Conversely, Lidar data lacks shadows but may omit certain buildings due to factors like reflective roof materials, humidity, or variables related to scanning angles, flight height, etc. Furthermore, while Lidar data may present buildings with zigzagged edges, it facilitates the recognition of planar features and offers an advantage in detecting small buildings within forested areas. Integrating imagery and Lidar data maximizes their strengths while minimizing their limitations.

The methods of building detection typically include unsupervised methods and supervised methods. Unsupervised methods typically utilize various features such as shapes, sizes, colors, texture patterns, etc. However, deep neural networks possess the remarkable ability to autonomously extract intricate, high-level features in a hierarchical fashion, leveraging massive training data. Supervised methods in this paper mainly focus on supervised deep neural networks. Most popular deep neural networks for building detection include VGG-16 (Simonyan & Zisserman, 2014), R-CNN (Girshick et al., 2014), DeepLab (Chen et al., 2014), UNet (Ronneberger et al., 2015), ResNet (He et al., 2016), YOLO (Redmon et al., 2016), etc. Thanks to the open sources, these neural networks were continuously developed as a family of networks, from VGG16 to VGG19, from R-CNN to Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2017) and its extension version: Mask R-CNN (He et al., 2017), from DeepLab to DeepLabv3+ (Chen et al., 2018), from UNet to UNet++ (Zhou et al., 2018) and UNet3+ (Huang et al., 2020), from YOLO to YOLOv8 (Jocher et al., 2023) and YOLO-v9 (Wang et al., 2024). ResNet was developed in five types of architectures with different numbers of neural network layers: ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152. Their performances were not proportional to the depths of the layers. ResNet50's performance has been widely acknowledged among these models. In recent years, vision transformers have been paid high attention due to their capability to capture global relationships in images, be flexible to the sizes of input data, and their architecture containing the self-attention mechanism. For such model architecture, there is potential for greater generalization to new data. Examples included the Cswin Transformer (Dong et al., 2022) and Omnivec (Srivastava & Sharma, 2024). These models outperformed the convolutionall neural networks (CNN). However, according to Moutik et al. (2023), the hybrid method with the combination of the CNN and vision transformers is more efficient and cost-effective. Regardless of neural network architectures, prediction accuracy and inference time are two important aspects in evaluating their performance. Nevertheless, model prediction accuracy relies on the model architecture and the types and quality of input data. The selection of the input data and the architecture of the model need to be considered in the practical applications.

3. Materials

Materials used in the project included the datasets of true orthophotos, LidarDTM, and LidarDSM, with 25cm spatial resolution, TDB building footprints, and a trained UNet++ model. The study areas covered twelve areas including southern, middle, and northern Finland. The experiment environment included urban and suburban areas (Oulu, Jyväskylä, Kuopio, Savonlinna, and Lahti in Finland), rural and forest areas (Uusikaarlepyy, Uusikaupunki, Nystad, Ylitornio, Parainen, Riihimäki, and Ylivieska in Finland). The datasets were stored in the NLS standard map sheet: 3000m x 3000m, with the format of GeoTIFF and the TDB building footprints were in .shp format. Python was the programming language. The trained UNet++ model was in .pth format. The model was run with PyTorch in the Spyder platform. GeoAI-derived building vectors were in a .shp format. QGIS was the open-sourced software used for quality control. The corrected TDB buildings were overlapped with true orthophotos and TDB vectors in the QGIS software to be inspected. The computations were carried out within the CSC, a high-performance computing environment provided by the Finnish IT Center for Science, owned by the Finnish state and higher education institutions (CSC, 2024). The CSC platform was used when the UNet++ model was trained with the new dataset. The new datasets were from the manually corrected GeoAI-derived building vectors. The correction was conducted by overlapping the building prediction with true orthophotos. Figure 1 shows the examples of datasets.

Figure 1. Examples of datasets.

From left to right: true orthophotos, DSM, DTM, GeoAI-derived building vectors.

4. Methods

Typically, there are five types of errors in geospatial data. They are related to positional accuracy, attribute accuracy, temporal accuracy, topological consistency, and data completeness. In the AI4DB project, the focus was on improving the positional accuracy and data completeness of TDB building vectors. The project leveraged the GeoAI model to identify buildings from the input data of true orthophotos, DTM, and DSM. The GeoAI-derived buildings served as reference data, enabling a comparison with building polygons from the TDB to recognize location deviations, missing structures, and demolished buildings. Fig. 2 shows the workflow of identifying the TDB building errors. The red text ‘Algorithms’ is described separately in Fig. 3.

The process began by generating true orthophotos and DSMs. This involved utilizing aerial images, their orientation data, waterbodies, and Lidar point clouds within SURE software (Esri) to produce high-resolution (25cm) true orthophotos. The subsequent step involved leveraging these true orthophotos, DTMs, and DSMs in the UNet++ model, for building prediction. The UNet++ model has been trained with high-quality datasets from a diversity of environments: urban, suburban, rural, or forest areas, covering one-third of Finland.

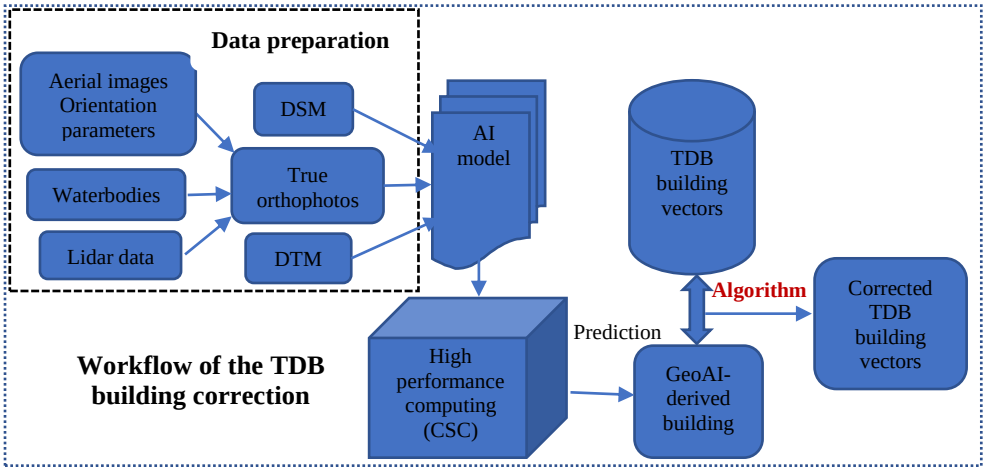

Figure 2. An overview of the workflow in the AI4TDB project

The accuracy of the predicted buildings was evaluated by the experts in the organization and it received up to 97% accuracy at the object level when compared to different types of reference data. In this article, the predicted buildings are also called 'GeoAI-derived buildings'. The GeoAI-derived vectors served as reference data for TDB building vectors, allowing the identification and rectification of location errors present within the TDB building vectors. The computations were carried out within the CSC.

4.1 Data preparation

While orthophotos stood as the standard products for the NLS, our choice leaned towards true orthophotos. In an orthophoto, buildings, and structures may appear skewed, but in a true orthophoto, photogrammetry software corrects these elements, presenting an accurate vertical perspective. Comparing orthophotos from different years might reveal discrepancies in building projections, whereas true orthophotos consistently maintain building locations regardless of the timeframe. True orthophotos can be generated from aerial images with orientation parameters, including water bodies, with or without Lidar data. The production of true orthophotos requires at least 30/60 aerial image overlaps.

4.2 GeoAI model prediction

The GeoAI model for building detection underwent its initial training phase within the scope of the ATMU project (the preceding AI project). The training dataset encompassed true orthophotos, building vectors, DSMs, and DTMs, all at a 25cm spatial resolution. Upon generating buildings from a GeoAI model that had been pre-trained, utilizing true orthophotos, DSMs, and DTMs was sufficient. These data were prepared in QGIS software: to select the same coverage of the area, to have the same spatial resolution, and to store them in GeoTIFF format, etc. Following data preparation, these data were input into the GeoAI model. The model's output consisted of detected buildings represented in a binary image format. To derive building vectors, a vectorization process was essential. However, these outlines initially retained numerous points. Implementing a building regularization process in ArcGIS Pro was necessary to streamline the building shapes and maintain their rectangular forms.

4.3 Accuracy control of GeoAI-derived buildings

Firstly, neither humans nor machines can achieve flawless results. Humans might make mistakes due to their carelessness or lack of experience. Machines might yield false detections, presenting false positives (indicating a detection where no building exists) or false negatives (failing to detect an existing building). The complexities causing false detection are multifaceted, with data quality and AI algorithms playing pivotal roles. Factors such as the radiation quality of aerial images, the accuracy of DSMs, and the resampling of DSMs and DTMs significantly impact the resulting accuracy. Maximizing result accuracy involves the collaboration between machines and humans. The detected building results underwent careful examination and rectification to ensure their accuracy. Within this project, building correction served dual purposes: ensuring accuracy for reference in correcting TDB building locations and retraining the GeoAI model.

4.4 To train the GeoAI model with corrected buildings

The quantity, quality, and diversity of training data significantly influence the predictive accuracy of the GeoAI model. Throughout the project, continuous enhancements were made to our GeoAI model. Following building correction, the model underwent training using all designated test areas. Test data in this project were organized in map sheets. Each map sheet covers 36 km^2. Most of the test areas were selected in full map sheets. At the beginning of the project, a few test areas were focused on the densely built areas, for example, the test data from Kuopio covered only 2km^2 and the Ylitornio test area covered 6km^2. Over the project duration, a total of 12 test areas, covering 2204km^2 (see Table 1), were trained in the model. To date, the GeoAI model has been trained using datasets from 35 production areas, covering approximately one-third of Finland.

4.5 To identify and correct the TDB building errors

'Algorithms' in Figure 2 are illustrated with details in Figure 3. It includes three functions: i) to identify and correct the TDB building polygon location errors, ii) to recognize the TDB missed buildings, and iii) to find the TDB demolished buildings. When the offset of TDB building locations was significant, it was corrected. If there was a rotation greater than three degrees between them, the TDB building polygon was rotated.

a) Input data:
Two sets of building vectors: GeoAI-derived buildings &

TDB buildings

b) Matching process:

- For each vector in the first set, the closest TDB vectors are searched.
- The intersection area is calculated for each closest TDB vector and the vector from the GeoAI-derived.
- The closest TDB vector is dropped if the intersection area is too small (10% of the GeoAI-derived polygon) or too large (97% of the AI-derived polygon),
- The closest TDB polygon is dropped because it might not represent the same building or the TDB polygon is either in a good position already or it is larger than the GeoAI-derived polygon.
- If there are multiple closest TDB polygons and the closest TDB polygons share a vector, the vectors are combined as one, otherwise, the polygons are dropped, because there is one GeoAI-derived polygon and multiple different TDB polygons.

c) Correct the TDB building polygon location errors
Spatial adjustment: Move the TDB building to minimize the area difference between the AI building and the TDB building and rotate the moved TDB building to achieve optimal alignment.

- The area sizes of the GeoAI-derived polygon and the closest TDB polygon are compared. If the difference between the areas is more than 25% of the GeoAI-derived polygon, the polygons are dropped because the polygons significantly differ from each other.
- The algorithm searches for the optimal position for the TDB polygon by maximizing the shared area and minimizing the area difference between the TDB polygon and the GeoAI-derived polygon and moving the TDB polygon to the optimal position.
- The TDB polygon is rotated to match the GeoAI-derived polygon by getting the minimum rotated rectangle of each polygon finding the corresponding sides of each rectangle with minimum angle difference and rotating the TDB polygon by that amount if the angle is larger than the minimum allowed rotation (3 degrees).
- The polygon is rotated and then, finally, the rotated TDB polygon and non-rotated TDB polygon are compared to the GeoAI-derived polygon. The rotation is kept if the rotated polygon intersects a larger area of the GeoAI-derived polygon than the non-rotated polygon.

d) Identify missed buildings:
One in the AI-derived building, but no corresponding building in the TDB

e) Recognize demolished buildings:
One in the TDB building, not in the AI-derived building

f) Quality assurance:
Assess the impact of the spatial adjustment between the AI and TDB buildings by checking the corrected TDB vectors with true orthophotos

g) Output:

- The corrected TDB polygons
- The distance that the TDB polygon moved
- Rotation angle
- Identified missed buildings
- Identified demolished buildings
- Coverage of the area
- Number of total buildings
- Number of municipality buildings
- Number of corrected municipality buildings
- Number of corrected non-municipality buildings
- Number of GeoAI-derived buildings
- The number of ignored TDB Polygons
- No Nearest TDB buildings
- Number of TDB buildings within the accuracy
- Number of one-to-many cases

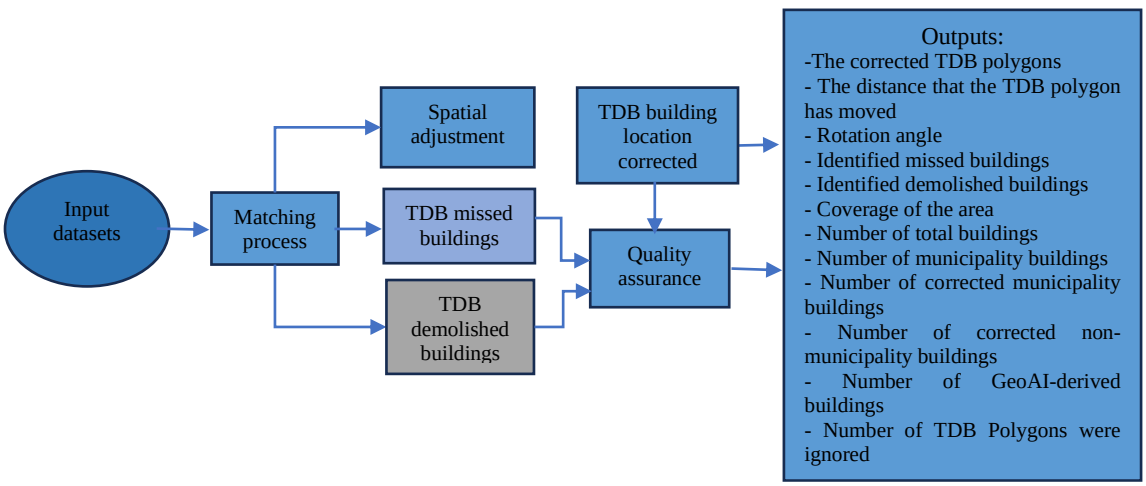

Figure 3. The algorithms include TDB building location correction, TDB missed and demolished buildings.

5. Results and evaluation

Throughout the project, datasets from twelve regions have been tested for identifying and correcting the TDB building locations including moving and rotation operations, covering a total area of 2204 km^2 (Table 2). Additionally, within the area of 1008 km^2, the missed and demolished buildings were recognized from the TDB vectors. The mapping team in the organization played the role of examining the results and giving feedback so that the algorithms could be continuously improved.

For the final assessment, five areas—Riihimäki (Southern Finland, 144 km^2), Jyväskylä1 (Middle of Finland, 144 km^2),

Jyväskylä2 (Middle of Finland, 288 km^2), Oulu (Northern Finland, 144 km^2), and Ylivieska (Middle of Finland, 288km^2) —were conducted. Overall information can be obtained from Table 1. In the Riihimäki area, 1104 out of 11333 building locations were corrected. 175 missed buildings and 1003 demolished buildings were recognized. In the Oulu area, 2041 out of 24276 buildings were corrected. 209 missed buildings and 3261 demolished buildings were identified. In the Jyväskylä1 area, 425 missed buildings and 2253 demolished buildings were found while 1929 out of 22081 buildings were corrected. The resulting percentages in Table 2 showed that urban areas with dense buildings appeared to have fewer errors when compared to the areas with a large coverage of forest. In Ylivieska, there were only 7701 buildings in an area of 288 km^2. The area was mostly covered by forests. The result was the worst among the others. When we compare Jyväskylä1 to Jyväskylä2, Jyväskylä1is located in urban area: 22081 buildings in an area of 144km^2. Errors were raised in rural or forest areas: 19.6% of the total buildings in Jyväskylä2 were assigned to the demolished buildings. With the GeoAI model, typically, forest areas encounter more false detection. TDB vector measurement from stereo images in forest areas might also be a challenge. Additionally, image quality, operator's experience, etc. affected the measurement.

Fig. 4 shows an example of the results in Jyväskylä test area. The left one demonstrated the GeoAI-derived buildings on the true orthophoto. The following one was the TDB missed buildings marked as Magenta. Then the next exhibits the differences between GeoAI-derived buildings and TDB buildings. The last one was the demolished buildings. It indicates that from the latest aerial image, the buildings did not exist, but they remained in TDB vectors.

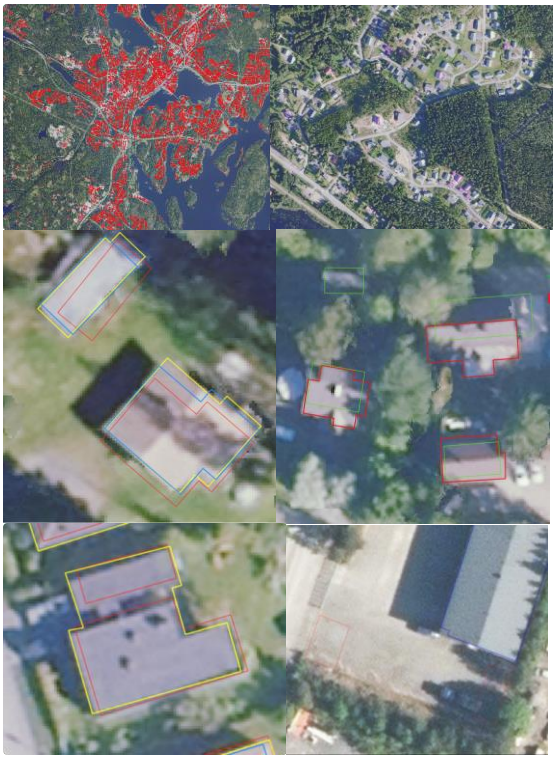

Figure 4. The test results were in Jyväskylä, Finland.
From left to right, up to down: a). GeoAI-derived buildings (Red), b). TDB missed buildings (magenta), c). buildings with location correction (red: TDB, yellow: AI, blue: corrected one), d). TDB buildings with location error (red: AI, green: TDB), e). Special case (AI vs. TDB): one polygon corresponding to two polygons, f). Demolished buildings (red)

Area Name	Oulu	Jyväskylä1	Riihimäki	Jyväskylä2	Ylivieska
Coverage (km^2)	144	144	144	288	288
Mapsheets	R4414L	N4324L	L4232R	N4323	Q4233
Total buildings: TDB	28642	22081	11546	11840	7701
Number of Missed Buildings	209	425	175	581	271
Number of Demolished Buildings	3261	2253	1003	2317	2291
Percentage of Missed Buildings	0.7%	1.9%	1.5%	4.9%	3.5%
Percentage of Demolished Buildings	11.4%	10.2%	8.7%	19.6%	29.7%

Table 1. A summary of the TDB missed and demolished buildings in test areas.

Name of area	**Map sheet**	**Area (km^2)**
Kuopio	P5114D	2
Savolinna	N5311A	54
Lahti	L4424	288
Vaala	R4334	288
Uusikaarlepyy	P3442	288
Uusikaupunki Nystad	Q3331EFG	54
Ylitornio	T4114H	6
Parainen	L3323L	216
Riihimäki	L4232R	144
Jyväskylä	N4324L, N4323	432
Oulu	R4414L	144
Ylivieska	Q4233	288

Table 2. The test areas for the TDB building vector location correction

In addition, essential information of each test area was estimated, including 'area', 'number of buildings', 'corrected buildings', 'one-to-one moved/rotated cases', 'buildings with significant differences in polygon areas', 'absence of nearest TDB buildings', 'one-to-many cases', etc. Such information offers insights into the overall status of TDB buildings and aids future decision-making processes.

Evaluation of the results was conducted by the mapping team in the organization. TDB building vectors were overlapped with the true orthophotos from the year 2023. Feedbacks from the sampled areas were delivered, mainly including i) the results were evidenced to be very useful, especially for those poor-quality areas, where good hints were provided; ii) Errors in the TDB building geometry or the AI prediction might cause the problems in the process of matching and correction. For instance, there are two separate buildings predicted by the GeoAI model while only one building polygon in the TDB data, or the other way around. iii) Small buildings surrounded by

trees might be missing from the AI detection; iv) Temporary buildings in construction areas might be detected as buildings; v) Underground buildings might be missed from the AI prediction; vi) Overall, GeoAI-derived buildings were beneficial for the TDB missed buildings and demolished buildings. However, in some cases, the demolished buildings after aerial imaging time (being removed from the TDB vectors) might be detected as missed buildings.

Based on the feedback received, it can be concluded that the use of AI methods was beneficial for improving data accuracy. However, some issues need humans to interact with. The collaboration between AI and humans will be the solution for the future.

6. Challenges

Throughout the AI4TDB project, we conducted a thorough analysis of the TDB building vectors and tested the functionality of the GeoAI model in real-world applications. From an AI methodology perspective, numerous factors influenced the model's predictions, including the types and quality of data sources, the environmental context (urban or forested areas), and the GeoAI model's robustness and adaptability to new datasets. The GeoAI model used in this project was the UNet++. With the rapid development of deep neural networks, more and more new deep network architectures surpassed the previous ones. An updating of the UNet++ model is needed. For example, in recent years, visual transformers with attention mechanisms outperformed CNN and became popular. The hybrid models of CNN and vision transformers are even better.

In addition, there existed a difference between GeoAI-derived building polygons and TDB-building polygons. The GeoAI-derived building polygons were obtained from true orthophotos, which removed building tilting, in a nadir-view. These building vectors were delineated from the building roofs while the TDB building footprints were measured either from the bases of buildings or the roofs according to the visibility from the aerial image stereo models. The differences measured from the building roofs and the bases were obvious. Some terraces were typically hanging on the top, joined as a part of roofs but not the part of the building bases. It makes the task even more challenging. These details have been considered during the algorithm development.

Furthermore, in the project, we employed intelligent strategies, yet a significant portion of the work necessitated manual intervention in quality assurance. To ensure result accuracy, the two-phase quality control process was implemented. All GeoAI-derived buildings underwent inspection against true orthophotos. Following the algorithm's identification of relocated or rotated buildings, these alterations were cross-checked against the TDB building vectors. Furthermore, GeoAI-derived buildings underwent manual correction and were utilized to train the GeoAI model.

Some issues with the GeoAI method were revealed by the mapping teams in the organization. For instance, temporary buildings in construction areas were detected as buildings, underground buildings might be missed from the AI prediction, the image cropping edges might cause problems in AI detection, etc.

7. Lessons

During the project duration, numerous tests were conducted to enhance building detection methodologies. For instance, we replaced aerial images' DSM with Lidar DSM and exclusively trained a distinct 3D UNet model utilizing Lidar data. Direct utilization of a 25cm DTM derived from Lidar data replaced the resamples from 2m DTM. Valuable insights were gained from these experiments. Notably, in forested areas, the AI model using true orthophotos and image DSMs proved suboptimal as it failed to detect many small buildings. Improved results were observed with the implementation of Lidar DSM. Additionally, the advantages of employing a 0.25m LidarDTM were evident.

However, relying solely on Lidar data for the 3D UNet model presented challenges as its strengths and weaknesses neutralized each other. The Lidar data's limitations, such as missing buildings due to strong reflections or high moisture levels on roofs, affected the outcomes. From these tests, a conclusive insight emerged—leveraging both true orthophotos and Lidar data presents the most viable solution for future endeavors.

8. Summary

The AI4TDB project showcased the practical application of GeoAI, specifically highlighting its effectiveness in utilizing the UNet++ with a set of input data consisting of true orthophotos, DSMs, and DTMs for predicting building vectors. These GeoAI-derived buildings were employed as reference data to be compared with the TDB building vectors. The errors of the TDB buildings in location deviation, structure missing, and redundancy were identified and corrected by the proposed algorithms. The experiment has been conducted in 2204 km^2 coverage with twelve test areas. During the experiment, the algorithm was continuously improved.

Final evaluation was carried out in five areas in Finland: Jyväskylä1, Jyväskylä2, Oulu, Riihimäki, and Ylivieska, with 1008km^2 in total. For instance, in the Jyväskylä1 area, the evaluation revealed 425 missed buildings and 2253 demolished structures, with corrections applied to 1929 out of 22081 buildings. The results revealed that urban areas emerged with fewer errors than rural or forest areas. The reason for this might be from more false detections appearing in forest areas, the data quality, the visibility of buildings on images, etc. Nevertheless, the project garnered positive feedback from the organization's mapping teams, endorsing its effectiveness in improving data accuracy. Mappers found value in the project as it significantly reduced manual workload by utilizing GeoAI-derived buildings as guiding references.

Despite encountering challenges, the project proved instrumental in garnering valuable insights and lessons for the future. Emphasizing a collaborative approach between humans and AI, the project advocates for a symbiotic relationship in production processes. Looking ahead, this collaborative model is posited as the optimal solution, acknowledging the strengths of both human expertise and AI capabilities.

Acknowledgment

The work was supported by the National Land Survey's AI4TDB project (Artificial Intelligence for Topographic Database Accuracy Enhancement), which was funded by the Ministry of Agriculture and Forestry in Finland for the period of 1.1.2023-31.12.2023.

The authors also wish to thank CSC - IT Center for Science, Finland (urn:nbn:fi:research-infras-2016072531) and the Open Geospatial Information Infrastructure for Research (Geoportti, urn:nbn:fi:research-infras-2016072513) for computational resources and support.

References

Boonpook, W., Tan, Y., Ye, Y., Torteeka, P., Torsri, K., & Dong, S. (2018). A deep learning approach on building

detection from unmanned aerial vehicle-based images in riverbank monitoring. *Sensors*, 18(11), 3921.

Chaloeivoot, T., & Phiphobmongkol, S. (2016). Building detection from terrestrial images. *Journal of Image and Graphics*, 4(1), 46-50.

Buyukdemircioglu, M., Can, R., Kocaman, S., & Kada, M. (2022). Deep learning based building footprint extraction from very high resolution true orthophotos and nDSM. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2, 211-218.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062.

Chen, R., Li, X., & Li, J. (2018). Object-based features for house detection from RGB high-resolution images. *Remote Sensing*, 10(3), 451.

CSC, 2024. www.csc.fi Accessed on March 12, 2024.

Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., ... & Guo, B. (2022). Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12124-12134).

Dornaika, F., Moujahid, A., El Merabet, Y., & Ruichek, Y. (2016). Building detection from orthophotos using a machine learning approach: An empirical study on image segmentation and descriptors. *Expert Systems with Applications*, 58, 130-142.

Ghasemian, N., Wang, J., & Reza Najafi, M. (2022). Building detection using a dense attention network from LiDAR and image data. *Geomatica*, 75(4), 209-236.

Gilani, S. A. N., Awrangjeb, M., & Lu, G. (2016). An automatic building extraction and regularisation technique using lidar point cloud data and orthoimage. *Remote Sensing*, 8(3), 258.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. "Rich feature hierarchies for accurate object detection and semantic segmentation", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 580, no. 587.

Ham, S., Oh, Y., Choi, K., & Lee, I. (2018). Semantic segmentation and unregistered building detection from UAV images using a deconvolutional network. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 419-424.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., ... & Wu, J. (2020, May). Unet 3+: A full-scale connected unet for medical image segmentation. In ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 1055-1059). IEEE.

Janowicz, K., Gao, S., McKenzie, G., Hu, Y., & Bhaduri, B. (2020). GeoAI: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. *International Journal of Geographical Information Science*, 34(4), 625-636.

Jiang, H., Li, Q., Jiao, Q., Wang, X., & Wu, L. (2018). Extraction of wall cracks on earthquake-damaged buildings based on TLS point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(9), 3088-3096.

Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. GitHub. 1 January 2023. Available online: https://github.com/ultralytics/ultralytics

Li, J., Huang, X., Tu, L., Zhang, T., & Wang, L. (2022). A review of building detection from very high resolution optical remote sensing images. *GIScience & Remote Sensing*, 59(1), 1199-1225.

Li, Y., Hu, Q., Wu, M., Liu, J., & Wu, X. (2016). Extraction and simplification of building façade pieces from mobile laser scanner point clouds for 3D street view services. *ISPRS international journal of geo-information*, 5(12), 231.

Liu, P., & Biljecki, F. (2022). A review of spatially-explicit GeoAI applications in Urban Geography. *International Journal of Applied Earth Observation and Geoinformation*, 112, 102936.

Malihi, S., Valadan Zoej, M. J., Hahn, M., Mokhtarzade, M., & Arefi, H. (2016). 3D building reconstruction using dense photogrammetric point cloud. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41, 71-74.

Moutik, O., Sekkat, H., Tigani, S., Chehri, A., Saadane, R., Tchakoucht, T. A., & Paul, A. (2023). Convolutional neural networks or vision transformers: Who will win the race for action recognitions in visual data? *Sensors*, 23(2), 734.
CSC, 2024. www.csc.fi Accessed on March 12, 2024

Nahhas, F. H., Shafri, H. Z., Sameen, M. I., Pradhan, B., & Mansor, S. (2018). Deep learning approach for building detection using lidar–orthophoto fusion. *Journal of sensors*, 2018.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Ren, S., He, K., Girshick, R. and Sun, J., 2017. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). 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 (pp. 234-241). Springer International Publishing

Saito, S., & Aoki, Y. (2015, February). Building and road detection from large aerial imagery. In *Image Processing: Machine Vision Applications VIII* (Vol. 9405, pp. 153-164). SPIE.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Sirko, W., Kashubin, S., Ritter, M., Annkah, A., Bouchareb, Y. S. E., Dauphin, Y., ... & Quinn, J. (2021). Continental-scale building detection from high resolution satellite imagery. arXiv preprint arXiv:2107.12283.

Srivastava, S., & Sharma, G. (2024). Omnivec: Learning robust representations with cross modal sharing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1236-1248).

Sun, S., & Salvaggio, C. (2013). Aerial 3D building detection and modeling from airborne LiDAR point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(3), 1440-1449.

VoPham, T., Hart, J. E., Laden, F., & Chiang, Y. Y. (2018). Emerging trends in geospatial artificial intelligence (geoAI): potential applications for environmental epidemiology. *Environmental Health*, 17, 1-6.

Wang, C. Y., Yeh, I. H., & Liao, H. Y. M. (2024). YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. arXiv preprint arXiv:2402.13616.

Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, *Proceedings 4* (pp. 3-11). Springer International Publishing.