GEOMATICS AND ENVIRONMENTAL ENGINEERING • Volume 20 • Number 6 • 2026

https://doi.org/10.7494/geom.2026.20.6.5

AI-Driven Building Detection for Updating Spatial Databases and Identifying Illegal Construction

Mateusz Czyrzniak*, Jacek Rapiński

University of Warmia and Mazury in Olsztyn, Olsztyn, Poland
* Corresponding author

Abstract: Possessing updated information on the spatial location of buildings is crucial in many surveying and urban planning processes. Updating databases using traditional surveying methods can be time-consuming and complicated by terrain conditions. Recent studies have demonstrated the potential of artificial intelligence and deep learning algorithms for identifying buildings from remote sensing data. Inspired by these innovations, this study proposes a practical tool for building detection from orthophotomaps. The paper presents a custom QGIS plugin designed for identifying buildings that are missing from existing spatial databases. The plugin integrates three deep learning models: Mask R-CNN, DeepLabV3, and YOLO11x-obb, each offering distinct advantages. YOLO11x-obb achieves the highest overall detection performance but outputs only oriented bounding boxes. In contrast, Mask R-CNN and DeepLabV3 provide detailed object boundaries. Mask R-CNN generally delivers better results, but DeepLabV3 can outperform it in scenarios requiring high pixel-level accuracy. The deep learning models were trained using a dataset of 273,644 building objects and 313 orthophoto images acquired from Geoportal (geoportal.gov.pl). Experimental results confirm the effectiveness of the proposed plugin in detecting unregistered buildings, highlighting its potential application for updating spatial databases, revising maps, and identifying previously undocumented constructions.

Keywords: object detection, deep learning, QGIS, orthophotomaps, buildings

Received: March 27, 2026; accepted: July 22, 2026

© 2026 Author(s). This is an open-access publication that can be used, distributed, and reproduced in any medium according to the Creative Commons Attribution 4.0 International License (CC BY 4.0).

E-mails and ORCID iDs: mateusz.czyrzniak@uwm.edu.pl, https://orcid.org/0009-0009-8968-2792 (M.C.); jacek.rapinski@uwm.edu.pl, https://orcid.org/0000-0002-8954-7963 (J.R.).

1. Introduction

Up-to-date spatial information regarding buildings is foundational to diverse urban processes, including urban planning, population estimation, property tax assessment, and public safety management. Accurate geospatial data enables governmental bodies to implement equitable property taxation and monitor land-use dynamics effectively. Furthermore, precise spatial data is critical for emergency response operations, evacuation planning, disaster risk mitigation, and continuous infrastructure monitoring [1].

Currently, the most commonly used techniques for acquiring spatial information about buildings include photogrammetry and remote sensing, while direct measurements are employed when high accuracy is required. However, direct measurements are notably time-intensive and cost-prohibitive, particularly in large-scale or rapidly developing urban environments. Although photogrammetry and remote sensing enable faster data acquisition, they often necessitate significant post-processing efforts. In recent years, there has been a paradigm shift toward automated solutions. Numerous studies have integrated advanced data acquisition techniques, such as unnamed aerial vehicles (UAVs), satellite imagery, and light detection and ranging (LiDAR), with artificial intelligence (AI) algorithms, specifically machine learning (ML) [2] and deep learning (DL) [3], to automate building extraction [4]. Despite these advancements, it is essential to note that many proposed methodologies remain at an experimental stage and have yet to be widely integrated into standard geographic information system (GIS) software workflows [5].

Traditional and still-popular DL architectures used for automated building detection primarily rely on convolutional neural networks (CNNs). Numerous studies have demonstrated the efficacy of these methods in spatial data extraction. One of the seminal models in this domain is U-Net [6], a framework designed for semantic segmentation. For instance, Xu et al. [7] employed U-Net for building extraction from very-high-resolution (VHR) remote sensing imagery, achieving an F1-score beyond 90%. Another widely utilized semantic segmentation model is DeepLabv3+. Wang et al. [8] reported pixel-level classification accuracy ranging from 85.53% to 97.79%, depending on the dataset, while their proposed refinement, MR-DeepLabv3+, further improved performance metrics by approximately 1%. Beyond semantic segmentation, more advanced CNN architectures, such as instance segmentation models, have gained prominence. Mask R-CNN [9] represents the state-of-the-art among CNN architectures. It performs extraction through a two-stage process: initial object detection followed by pixel-level segmentation of the identified regions. Utilizing this architecture on aerial imagery acquired by UAVs, Kaya [10] achieved an F1-score of 95.42%.

Recent research has increasingly focused on novel architectures based on transformers, which offer superior capabilities in capturing long-range dependencies and global contextual information. These models utilize self-attention mechanisms,

enabling them to effectively mitigate challenges such as local illumination variations within imagery [11]. A prominent model in this category is Mask2Former. Gibril et al. [12] evaluated the performance of transformer-based architectures against CNN-based benchmarks for building extraction. Mask2Former achieved an F1-score of 94.85%, demonstrating an improvement over the 94.42% recorded by DeepLabv3+. Furthermore, the Segment Anything Model (SAM) [13] has emerged as a significant advancement in foundational computer vision. Unlike traditional supervised models, SAM exhibits remarkable zero-shot generalization capabilities, allowing it to extract objects not encountered during its training phase. However, its effectiveness typically relies on user-provided prompts, such as points or bounding boxes. To address this limitation, Zhang et al. [14] proposed an automated approach by integrating two specific modules into a vision transformer (ViT) architecture. An "Adapter-Scale" module was introduced to perform dimension reduction, GELU activation, and dimension restoration, while an "Adapter-Feature" module was embedded between ViT layers to extract high-frequency features, which serve as intrinsic prompts for subsequent layers. While the standard SAM required manual prompting to achieve an F1-score of 79.65% for building extraction, the proposed automated adaptation successfully reached 83.37%.

Despite the aforementioned advancements, the availability of robust, user-accessible tools for automated building extraction remains constrained. Within the ArcGIS ecosystem, Esri provides a pre-trained Mask R-CNN model that supports fine-tuning on custom datasets; however, this solution is proprietary and imposes substantial computational requirements during the training phase. Conversely, the QGIS platform offers community-driven alternatives via various plugins. For instance, the Deepness plugin facilitates access to a building segmentation model. While it achieves a competitive F1-score of 81.0% on its validation set, its utility is restricted by several factors. Specifically, the model is trained on a relatively small dataset (3,468 samples) and geographically constrained to a single urban area (Paris), which significantly impairs its generalization capabilities across geographically diverse environments. Furthermore, the fixed input resolution of 256 × 256 pixels may lead to the omission of larger building footprints or structural segments. Other QGIS plugins, such as Mapflow, also support automated building detection but, similar to the ArcGIS suite, operate under a commercial license. Recently, both ArcGIS and several QGIS plugins (e.g., GeoAI) have integrated the SAM, facilitating rapid object extraction through prompt-based interaction. Nevertheless, all these solutions primarily focus on the extraction phase, leaving subsequent tasks, such as multi-temporal change detection, to be performed by the user through additional, disparate analytical tools within the GIS environment.

The primary objective of this study is to develop an open-source, user-friendly QGIS plugin designed for automated building detection from orthophotomaps using DL methodologies. This tool is intended to streamline multi-temporal change detection by identifying buildings on current orthophotos and subsequently comparing

these findings with existing records in a spatial database. This automated approach significantly accelerates the detection of newly constructed structures, including those built without appropriate permits, which may circumvent building codes and compromise public safety. Furthermore, the proposed solution enables the identification of demolished structures, thereby facilitating more effective tax compliance and supporting urban planning initiatives by maintaining accurate, high-fidelity spatial datasets.

A significant challenge in extracting building footprints from aerial imagery arises from perspective and relief distortions inherent in orthophotos, which often compromise the boundary precision of DL inferences. In this study, ground-truth building boundaries used for training are sourced from Geoportal, provided by the Head Office of Geodesy and Cartography (geoportal.gov.pl). As these records typically represent the outlines of buildings at the ground level (footprint), rather than their roof structures as captured in overhead imagery, the training samples are inherently subject to spatial discrepancies. Rather than treating this issue as an intractable constraint, this study evaluates the efficacy of diverse architectural configurations in mitigating these spatial errors.

The primary scientific and engineering contributions of this study are:

1) a comprehensive evaluation of distinct DL paradigms to establish an optimal balance between geometric delineation, object detection, and inference efficiency;
2) implementation of an automated spatial analysis pipeline that performs topology-based comparisons between vectorized AI model predictions and existing official databases to isolate new man-made structural footprints.

The remainder of this paper is organized as follows: section two describes the methodology, section three presents the results of the AI models trained and the implementation of the QGIS plugin, and section four contains the final thoughts about the study.

2. Methodology

This section presents the methodology applied in this research, including the tools and workflows used to develop the proposed application and to validate its performance. It outlines the overall experimental setup, from data acquisition and preprocessing to model selection, training, and evaluation.

2.1. DL Models

The key element of the developed plugin is an AI algorithm; therefore, the selection of an appropriate model significantly affects the performance of the feature. In the context of remote sensing and automated mapping, building footprint extraction from high-resolution imagery can be addressed through distinct computer vision paradigms. To systematically evaluate which methodology best addresses the practical requirements of maintaining a spatial database and identifying illegal

constructions, this study compares three primary architectural paradigms: instance segmentation [15], semantic segmentation [16], and object detection [17].

In the model selection process, particular consideration was given to the geometric displacements inherent in aerial imagery, which can adversely affect the convergence and precision of training procedures. While state-of-the-art transformer-based architectures often demonstrate superior performance in capturing intricate structural details, we hypothesized that their global capacity might render them more susceptible to the adverse effects of spatial discrepancies and noise within training datasets. This design assumption aligns with recent findings by Khanal et al. [18], who demonstrated that vision transformers trained from scratch exhibit a heightened susceptibility to overfitting on imperfect or noisy training data compared to convolutional counterparts. Furthermore, as demonstrated by Gibril et al. [12], CNN-based models remain highly competitive for building recognition tasks. Consequently, to ensure greater robustness against potential spatial discrepancies, we selected architectures based on CNNs.

Instance segmentation is frequently employed for building footprint extraction, as it provides an end-to-end framework capable of performing concurrent object detection and pixel-level instance delineation. The resulting output, in the form of a discrete instance mask, is particularly advantageous, as it facilitates direct comparison with existing records in spatial databases. However, these architectures are notably computationally intensive. For this study, the Mask R-CNN [9] was selected due to its proven efficacy in remote sensing workflows [19]. Beyond its performance, the industrial validity of Mask R-CNN is underscored by its native implementation within major commercial GIS platforms, such as Esri's ArcGIS, establishing it as a primary benchmark for state-of-the-art building extraction.

Conversely, semantic segmentation reduces computational overhead by performing dense, pixel-level classification without the need for complex instance differentiation. This approach typically achieves superior pixel-level accuracy in delineating intricate boundaries. Unlike instance segmentation, which relies on object proposals, semantic segmentation also evaluates the entire image, thereby minimizing the risk of omitting individual structures. While buildings that share boundaries cannot be partitioned into distinct instances during the conversion of masks to vector formats, such separation is not strictly required for multi-temporal change detection. Identifying the spatial overlap between classified building regions and existing database records is often sufficient. DeepLabV3 [20] was selected as a representative semantic segmentation model. This architecture performs dense, pixel-level classification and has proven highly effective in a variety of remote sensing and urban mapping applications [21]. DeepLabV3+ was also considered a suitable candidate [22], but it was ultimately excluded from the final experimental suite. This decision was treated as a design assumption based on the structural differences in the decoder mechanisms of the two architectures. DeepLabV3+ incorporates a dedicated decoder module designed to leverage low-level features for sharp boundary

recovery. However, we hypothesized that this heightened sensitivity to fine-grained features might exacerbate the impact of geometric discrepancies and labeling inconsistencies inherent in our training datasets by inducing localized overfitting to annotation errors along object contours.

Finally, the object detection paradigm is characterized by high computational efficiency and rapid inference speeds. Unlike segmentation models, these architectures do not perform pixel-level classification; instead, they identify objects by predicting bounding boxes. Nevertheless, if the primary objective is the rapid localization of newly constructed objects rather than high-fidelity boundary extraction, pixel-level segmentation may not be strictly required, making oriented object detection a more efficient alternative. On the other hand, this approach introduces potential discrepancies when integrating results with existing spatial databases, as bounding boxes often include significant areas of the surrounding environment. If the shape of the building is not rectangular (e.g. it is L-shaped) the area of the bounding box may include other structures, thereby distorting the comparison with the database. In this study, the YOLO11x-obb version was used. The "obb" acronym stands for "oriented bounding boxes." By producing angled bounding boxes that more tightly conform to the actual orientation and footprint of the structure, the area of the bounding boxes is reduced, but this does not solve the problem entirely (Fig. 1).

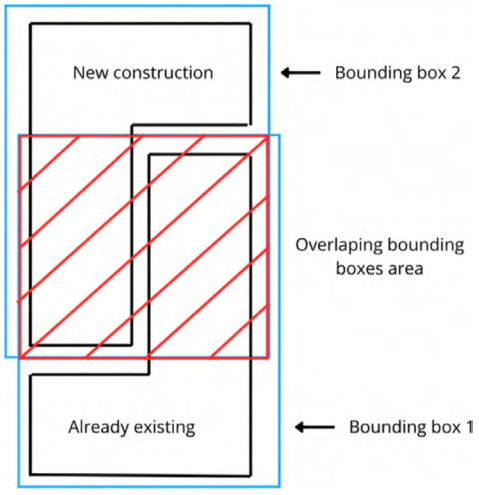

Fig. 1. Conceptual illustration of the geometric discrepancies introduced by bounding boxes for non-rectangular structures (e.g., L-shaped footprints)

2.2. Data Collection

Training an AI model requires a substantial amount of data. Manual data acquisition, involving drone flight campaigns and exhaustive manual labeling, would be extremely time-consuming. To address this limitation, the research community has

developed numerous open-source databases containing building footprints. However, because building architecture varies significantly between countries, and since the primary functional scope of the plugin is focused on Poland, building footprint data was obtained free of charge from Geoportal (geoportal.gov.pl), which is a government-maintained spatial information system for Poland. In total, 273,644 building instances and 313 orthophoto images were collected from diverse areas, including villages, urban environments, and agricultural regions. The images had a resolution of approximately 10,000 × 10,000 pixels and a ground sampling distance (GSD) of 0.25 m [23]. In remote sensing, the GSD defines the physical distance on the ground represented by the linear dimension of a single pixel. This parameter fundamentally governs the spatial resolution of the dataset, dictating the level of geometric detail accessible to the computer vision models, thereby affecting their accuracy.

This approach, while efficient, has certain limitations. The appearance of buildings in orthophotos is affected by perspective distortion caused by the camera projection center, whereas building footprints are mapped at ground level. As a result, a spatial offset is introduced between the labels and the imagery (Fig. 2). Another limitation arises from data incompatibility. The building footprint layers and orthophoto imagery are not updated simultaneously, which may lead to inconsistencies between the datasets. As a result, there are instances where buildings are visible in the imagery but have not yet been recorded in the building database. There are also opposite cases where buildings are present in the database but no longer exist.



Fig. 2. Visual example of collected data, burdened with spatial misalignment

The resolution of the collected images was too high for efficient training of the AI models. Training on images with a resolution of 10,000 × 10,000 pixels would require substantial computational resources. Therefore, the images were divided into smaller tiles and subsequently padded to a uniform resolution of 1,000 × 1,000 pixels, as

the models require input images of fixed dimensions. For each generated tile, a corresponding shapefile containing the relevant building annotations was created. The data were divided into three subsets: training, validation, and test datasets (Table 1).

Table 1. Distribution of data across subsets

Metric	Training	Validation	Test	Total
Number of buildings	223,126	33,741	16,777	273,644
Number of tiles	8,132	1,321	662	10,115
Number of original images	139	87	87	313

2.3. Model Evaluation

The models were evaluated using the following metrics: precision, recall, F1-score, and mean Intersection over Union (mIoU).

Precision (1) – percentage of correct predictions from all predictions made by the model [24]:

$$Precision = \frac{T_P}{T_P + F_P} \tag{1}$$

where: T_P – true positive, and F_P – false positive.

Recall (2) – percentage of detected objects from all objects [25]:

$$Recall = \frac{T_P}{T_P + F_N} \tag{2}$$

where: T_P – true positive, and F_P – false positive.

F1-score (3) – overall score of the model based on precision and recall [26]:

$$F1 - score = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

Mean IoU (4) – mean percentage of overlap between prediction and ground truth building footprint [27]:

$$mIoU = \frac{1}{n}\sum_{i=1}^{n} IoU_i \tag{4}$$

$$IoU_i = \frac{\left|A_i \cap B_i\right|}{\left|A_i \cup B_i\right|} \tag{5}$$

where: A – prediction area, B – ground truth area, and n – number of samples.

2.4. Plugin Setup

QGIS is a free and open-source GIS application used for map creation and spatial data analysis. It provides all essential tools and functionalities, including support for shapefiles, orthophotomaps, and coordinate reference systems [28]. QGIS also offers a plugin framework that allows users to develop and integrate custom solutions. Plugin Builder 3 [29] was used to develop the proposed plugin. This tool enables the rapid setup of a complete code environment for plugin development.

The plugin workflow is illustrated in Figure 3 and consists of the following steps: selecting an AI model, setting the detection parameters, defining the area of interest, performing the detection, and comparing the resulting shapefile layer with a reference layer. The user interface is shown in Figure 4.

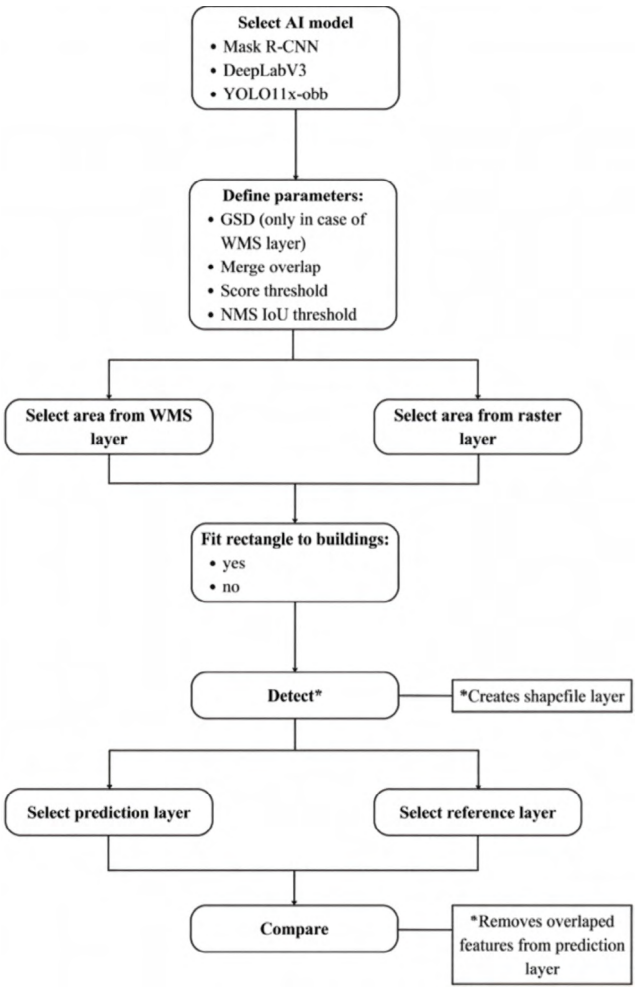

Fig. 3. Application workflow

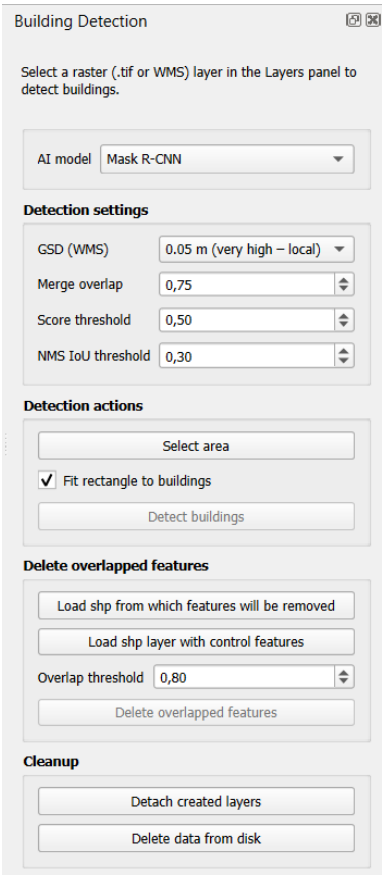

Fig. 4. QGIS plugin user interface

First, the user is asked to select one of three AI models: Mask R-CNN, DeepLabV3, or YOLO11x-obb. The second step involves selecting four parameters:

1) GSD value – this parameter is used only when performing detection from a Web Map Service (WMS) layer [30]. When a WMS layer is used, the software downloads the selected area as a.tif file, with the GSD determined by this parameter.
2) Merge overlap – as the model requires small input images and the selected area may be large, the image is divided into smaller tiles with a resolution of 1,000 × 1,000 pixels. As a result, some buildings may be split across tile boundaries, which can lead to detection issues. To address this problem, tiles were created with an overlap of 300 pixels. In the final step, detections from all tiles were merged into a single layer. This parameter prevents objects being duplicated and enables the merging of overlapping detections when a specified spatial overlap percentage threshold is reached.

3) Score threshold – this AI model parameter defines which detections are retained and which are discarded. Detections with a confidence score below this threshold are removed [31]. This parameter is applicable only to the YOLO11x-obb and Mask R-CNN models.
4) NMS (non-maximum suppression) IoU threshold – this AI model parameter is used to eliminate redundant or overlapping detections. When multiple detections correspond to the same building, the NMS algorithm compares their bounding boxes or masks and retains only the detection with the highest confidence score. Overlapping detections are suppressed if the IoU between them exceeds the specified threshold. Adjusting this parameter helps to reduce false positives [32]. This parameter is applicable only to the YOLO11x-obb and Mask R-CNN models.

In the third step, the user selects the area from a raster or WMS layer where the detection is performed. At this stage, the user can also choose whether to fit rectangular shapes to the predicted features (applicable only to Mask R-CNN and DeepLabV3). This option produces smoother, more regular-shaped predictions, as illustrated in Figure 5.

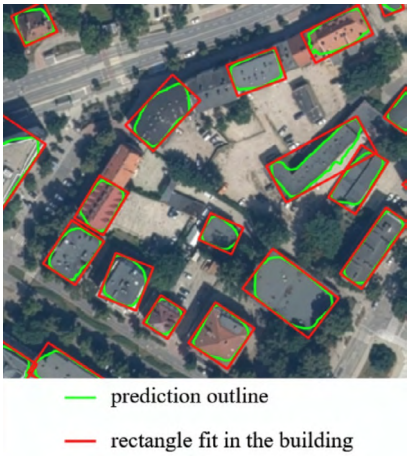

Fig. 5. Geometric regularization of building outline detection.
The green outline is the detection output. The red outline represents the smallest rectangle drawn around the outline

The detection process can be performed after the parameters and the area of interest have been selected. Once completed, a shapefile layer containing the detected buildings is added to the application. In the final step, the user is asked to select two shapefile layers: one with the detections and another as the reference layer. During this process, features in the detection layer that overlap with the reference layer above a specified threshold are removed. As a result, the final layer contains only features that are not present in the reference database.

3. Results

Each of the three models was trained up to fifteen times with different hyperparameter settings until no further improvement was observed. The behavior of the loss function was analyzed to determine the optimal hyperparameters (Fig. 6). Based on these observations, the number of training epochs was reduced, and the remaining hyperparameters were adjusted in subsequent iterations. Each model was then evaluated on the test dataset to verify whether the introduced changes led to improved performance. The configuration yielding the best results for each AI model is presented in Table 2. The computational experiments and model training were conducted on a high-performance Linux server (Ubuntu 24.04.4 LTS). The hardware environment consisted of an Intel Xeon Gold 6526Y CPU (16 cores, 32 threads), 128 GB of system RAM, and an Nvidia L40S GPU with 48 GB of VRAM.

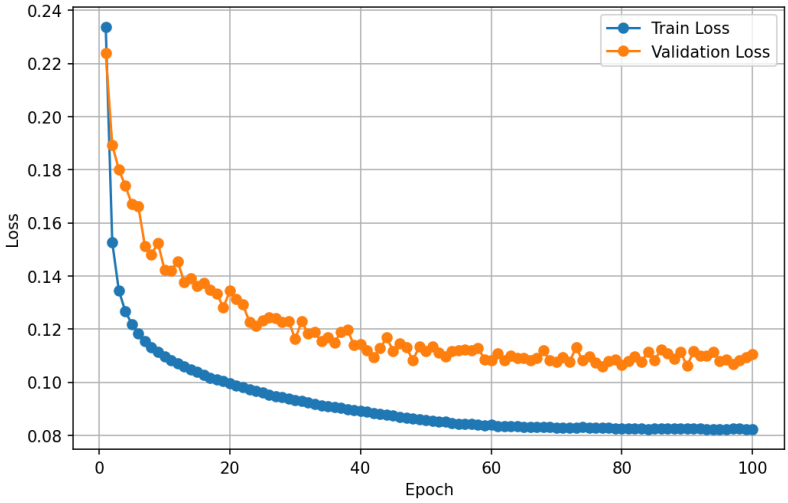

Fig. 6 Example of loss function for one of the trained DeepLabV3 models

Table 2. Best hyperparameter settings for each model

Hyperparameter	Mask R-CNN	DeepLabV3	YOLO11x-obb
Epochs	30	50	100
Batch size	2	2	8
Learning rate	0.001	0.001	0.010
Optimizer	SGD	AdamW	SGD
Weight decay	0.0005	0.0010	0.0005
Momentum (SGD)	0.900	–	0.937
β_1 (AdamW)	–	0.900	–

Different hyperparameter configurations yielded optimal performance depending on the model architecture. For Mask R-CNN and YOLO11x-obb, the best-performing configurations utilized the Stochastic Gradient Descent (SGD) optimizer [33], whereas AdamW [34] achieved better results for DeepLabV3. Regarding the learning rate and batch size, Mask R-CNN and DeepLabV3 employed lower values of 0.001 and 2, respectively. As the SGD optimizer incorporates momentum, its value for YOLO11x-obb was determined automatically by the Ultralytics algorithm, while it was manually tuned for Mask R-CNN. AdamW, which employs the β_1 parameter [35] instead of momentum, was also configured manually. Finally, the weight decay [36] value was identical for Mask R-CNN and YOLO11x-obb, whereas it was doubled for DeepLabV3.

Each model was evaluated on the test dataset using four metrics: precision, recall, F1-score, and mIoU. The evaluation was conducted across three Intersection over Union (IoU) thresholds: 0.75, 0.50, and 0.10. These thresholds determine the validity of a prediction: if the overlap between the predicted and reference polygons exceeds the specified threshold, the prediction is considered a true positive. The mIoU was calculated exclusively based on these positive detections. The results are summarized in Tables 3, 4, and 5. Table 6 contains detailed detection metrics such as true positive (TP), false positive (FP) and false negative (FN). A threshold level of 0.10 is generally not recommended for visual detection tasks due to its tendency to artificially inflate detection rates and increase the risk of FP detections. Nevertheless, the plugin is intended as a decision-support tool rather than a fully autonomous solution. Since final verification is always performed by a human operator, validating whether an AI-generated polygon represents a TP or a FP detection is considerably faster and less labor-intensive than manually inspecting extensive orthophotomap layers to identify structural discrepancies.

Table 3. Model results with IoU threshold 0.75

Model	Precision [%]	Recall [%]	F1-score [%]	mIoU [%]
Mask R-CNN	27.87	22.69	25.02	82.08
DeepLabV3	25.46	17.20	20.53	83.07
YOLO11x-obb	40.11	40.08	40.09	84.78

Table 4. Model results with IoU threshold 0.50

Model	Precision [%]	Recall [%]	F1-score [%]	mIoU [%]
Mask R-CNN	61.12	49.76	54.86	72.62
DeepLabV3	50.53	34.14	40.75	73.72
YOLO11x-obb	64.26	64.22	64.24	77.42

Table 5. Model results with IoU threshold 0.10

Model	Precision [%]	Recall [%]	F1-score [%]	mIoU [%]
Mask R-CNN	75.01	61.07	67.32	66.07
DeepLabV3	71.92	48.60	58.00	62.16
YOLO11x-obb	73.89	73.84	73.87	71.29

Table 6. Detailed detection result metrics

Model	IoU Threshold	True Positives	False Positives	False Negatives
Mask R-CNN	0.10	10,245	3,413	6,532
	0.50	8,348	5,310	8,429
	0.75	3,807	9,851	12,970
DeepLabV3	0.10	8,153	3,183	8,624
	0.50	5,728	5,608	11,049
	0.75	2,886	8,450	13,891
YOLO11x-obb	0.10	12,388	4,377	4,389
	0.50	10,774	5,991	6,003
	0.75	6,724	10,041	10,053

The results obtained at different IoU thresholds vary significantly. The highest precision, recall, and F1-score values are observed at the lowest IoU threshold of 0.10. Conversely, the mIoU metric exhibits the opposite trend: higher IoU thresholds yield higher mIoU values, reflecting a more precise spatial alignment between predicted polygons and ground-truth annotations.

The YOLO11x-obb model consistently achieves the best performance across nearly all evaluation metrics and IoU thresholds in the conducted experiments. The sole exception is precision at an IoU threshold of 0.10, where Mask R-CNN slightly outperforms YOLO11x-obb. It is important to note that YOLO11x-obb produces oriented bounding boxes rather than segmentation masks. Consequently, its

performance metrics are derived from comparisons between predicted and ground-truth bounding boxes, meaning its high metrics reflect geometric alignment within this specific output representation. Despite its high performance, bounding box detection can introduce inaccuracies when cross-referencing results with existing database records. For instance, the bounding box of a newly constructed building may overlap with neighboring objects, leading to potential discrepancies during layer comparison. Consequently, for YOLO11x-obb, utilizing low IoU thresholds when validating detections against database data is discouraged.

The magnitude of the performance differences between the IoU thresholds depends on the evaluated metric. For precision, recall, and F1-score, the largest gap is observed between the IoU thresholds of 0.50 and 0.75. At the strict threshold of 0.75, both DeepLabV3 and Mask R-CNN achieve relatively low scores of around 20%, while YOLO11x-obb reaches approximately 40%. When the IoU threshold is reduced to 0.50, the performance of DeepLabV3 and Mask R-CNN nearly doubles, with only a moderate decrease in mIoU of approximately 10%. The trade-off between IoU thresholds of 0.50 and 0.10 appears more balanced. Further relaxing the threshold leads to an additional increase of about 10% in precision, recall, and F1-score, accompanied by a comparable decrease in mIoU.

In terms of border estimation, Mask R-CNN generally outperforms DeepLabV3. The highest values of precision, recall, and F1-score across all evaluated IoU thresholds were achieved by Mask R-CNN at an IoU threshold of 0.10, reaching 75.01%, 61.07%, and 67.32%, respectively. In contrast, the highest mIoU value was obtained by DeepLabV3 at the strict IoU threshold of 0.75, indicating superior pixel-level boundary alignment under more demanding overlap conditions.

Analysis of Table 6 shows that reducing the IoU thresholds transfers many FP detections into TP detections. Based on these findings, it can be inferred that numerous detections at high IoU values are correct but spatially shifted relative to the ground-truth objects. Consequently, the IoU threshold criterion is not satisfied, and the detection is classified as a FP. Furthermore, as evidenced by this table, despite achieving the highest F1-score and the lowest number of FNs, the YOLO11x-obb model yields the highest number of FP detections.

The detection results are illustrated in Figures 7 and 8. These figures present two residential areas characterized by high and low building density, respectively. From Figure 7, it can be observed that the Mask R-CNN and YOLO11x-obb models perform moderately in detecting instances of adjacent structures. DeepLabV3 detected these structures, but it classified them as a single object. Because of this, the automatic verification of detections against the reference data remains problematic. In this case, confirming the existence of a building in the database would require the use of a low IoU threshold. In contrast, Figure 8 shows that all models perform well in detecting buildings that are clearly separated from one another. However, DeepLabV3 exhibits a tendency to merge neighboring instances when the distance between them is relatively small.

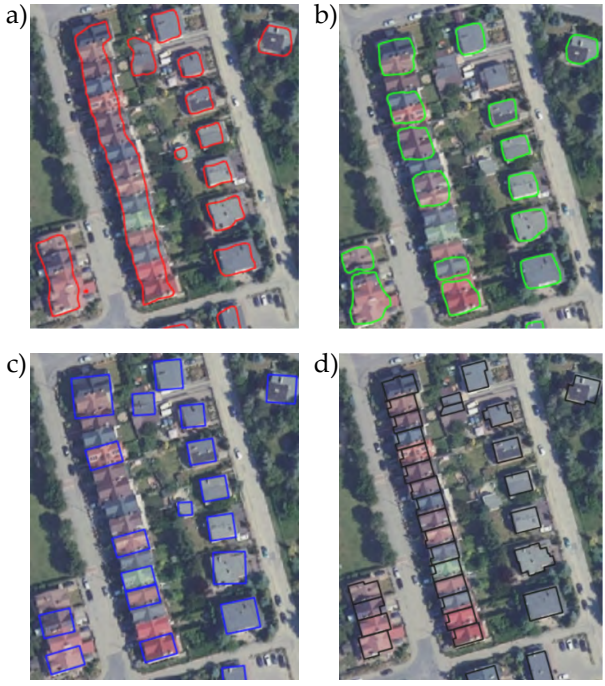

Fig. 7. Comparison of model detection results in a high-density residential area: a) DeepLabV3; b) Mask R-CNN; c) YOLO11x-obb; d) ground truth

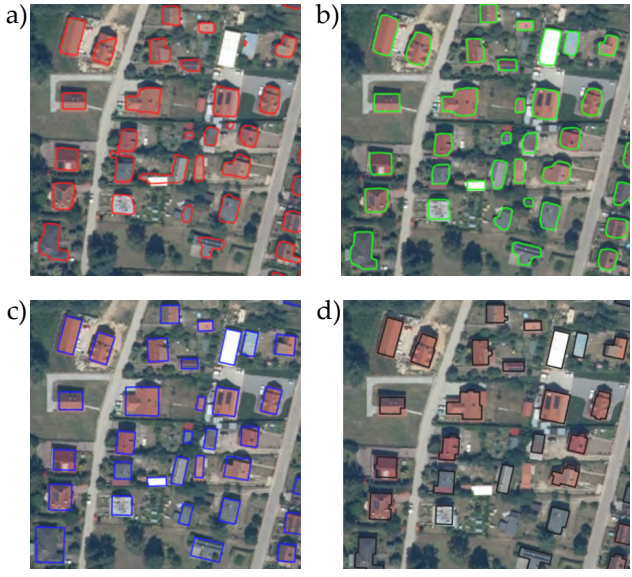

Fig. 8. Comparison of model detection results in a low-density residential area: a) DeepLabV3; b) Mask R-CNN; c) YOLO11x-obb; d) ground truth

4. Conclusions

Acquiring up-to-date geospatial information on buildings is crucial for many surveying and spatial analysis processes. This study presents a QGIS plugin designed for automated building detection from orthophotomaps using AI-based models. The plugin enables automated detection, the comparison of predictions with reference datasets, and the removal of overlapping objects, resulting in a final set containing only those buildings that are absent from the reference database.

The user can select one of three detection models: Mask R-CNN, DeepLabV3, or YOLO11x-obb. Several parameters can also be adjusted prior to detection, including the score threshold, merge overlap, and NMS IoU threshold. A fourth parameter, the GSD size, is used exclusively during raster acquisition from a WMS layer. The plugin also allows the user to define the overlap threshold applied when comparing predictions with reference data, providing flexibility in adapting the tool to various application requirements.

Depending on the use case, a different model may be preferred. Mask R-CNN and DeepLabV3 extract object boundaries, whereas YOLO11x-obb outputs oriented bounding boxes. For consistency, the plugin also offers the option to generate bounding boxes from boundary-based predictions. A model's performance varies significantly with respect to the required IoU overlap between predictions and ground truth. All models achieve better results at lower IoU thresholds, while performance decreases as stricter overlap criteria are applied. Overall, within the scope of this study, YOLO11x-obb achieves the highest overall metrics among the tested models, exceeding 70% in all evaluated criteria at an IoU threshold of 0.10. However, comparative analysis against the reference database is prone to errors in specific scenarios. This limitation stems from the inclusion of extraneous areas within the oriented bounding boxes, highlighting the inherent differences between box-based and pixel-level evaluations. Furthermore, under this experimental setup, YOLO11x-obb has the lowest number of FNs but also the highest number of FPs. Mask R-CNN generally ranks second in terms of the conducted benchmarks, outperforming DeepLabV3 in most cases. However, DeepLabV3 achieved a higher mIoU value for a high IoU threshold of 0.75, requiring precise boundary alignment.

The evaluated models demonstrate high utility in applications where precise boundary delineation is not required. When a high IoU threshold is applied, many otherwise correct detections are rejected due to insufficient overlap with the reference data. Lowering the IoU threshold results in a substantial number of FP detections being reclassified as TPs. Consequently, the developed plugin can serve as an effective decision-support tool for identifying newly constructed buildings in orthophotomaps. By filtering potential candidates from the set of detected objects, the proposed approach significantly accelerates the identification of new structures compared with exhaustive manual visual inspection. In contrast, the performance decreases considerably when accurate boundary delineation is required. At an IoU threshold of 0.75,

the precision, recall, and F1-score decline to approximately 20% for Mask R-CNN and DeepLabV3, and to around 40% for YOLO11x-obb. At the same time, the corresponding mIoU values exceed 80%, indicating that the remaining detections exhibit a high degree of spatial agreement with the reference data.

The reduced performance in precise boundary detection tasks is likely influenced by inaccuracies in the training data, particularly spatial misalignments between the building outlines visible in raster imagery and the corresponding vector boundaries obtained from Geoportal (geoportal.gov.pl). YOLO11x-obb may be influenced less by this problem because it is trained on oriented bounding boxes created based on vector boundaries of buildings. Despite these imperfections, the models successfully learned to recognize building locations and general shapes, demonstrating robustness to a certain level of annotation noise.

Future work should focus on retraining and fine-tuning the models using higher-quality, well-aligned datasets free from spatial shifts. Further development could include extending the plugin with additional models, incorporating post-processing techniques for boundary refinement, and performing broader validation across diverse geographic regions. Another promising approach may be to combine models, utilizing an object detection model first to locate buildings, followed by a segmentation model to extract precise boundaries. Such improvements could significantly enhance the tool's applicability for tasks requiring higher geometric accuracy while preserving its efficiency in large-scale building detection workflows.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CRediT Author Contribution

M.C.: conceptualization, methodology, resources, software, validation, writing – original draft, data curation, formal analysis, investigation, visualization, project administration.

J. R.: conceptualization, methodology, supervision, writing – review and editing.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Requests for access to the datasets generated or analyzed during this research will be considered upon inquiry to the corresponding author.

Use of Generative AI and AI-Assisted Technologies

Generative AI tools were used in the preparation of this manuscript. ChatGPT was used solely as a language editing tool to check English grammar and improve readability.

References

[1] Sirko W., Kashubin S., Ritter M., Annkah A., Bouchareb Y.S.E., Dauphin Y., Keysers D., Neumann M., Cissé M., Quinn J.: *Continental-scale building detection from high resolution satellite imagery*. arXiv preprint, arXiv:2107.12283, 2021. https://doi.org/10.48550/arXiv.2107.12283.

[2] Bell J.: *What is machine learning?*, [in:] Carta S. (ed.), *Machine Learning and the City: Applications in Architecture and Urban Design*, John Wiley & Sons, Hoboken 2022, pp. 207–216. https://doi.org/10.1002/9781119815075.ch18.

[3] Mathew A., Arul A., Sivakumari S.: *Deep learning techniques: An overview*, [in:] Hassanien A.E., Bhatnagar R. (eds.), *Advanced Machine Learning Technologies and Applications*, Springer, Singapore 2021, pp. 599–608. https://doi.org/10.1007/978-981-15-3383-9_54.

[4] Han Q., Yin Q., Zheng X., Chen Z.: *Remote sensing image building detection method based on Mask R-CNN*. Complex & Intelligent Systems, vol. 8(3), 2022, pp. 1847–1855. https://doi.org/10.1007/s40747-021-00322-z.

[5] Li J., Huang X., Tu L., Zhang T., Wang L.: *A review of building detection from very high resolution optical remote sensing images*. GIScience & Remote Sensing, vol. 59(1), 2022, pp. 1199–1225. https://doi.org/10.1080/15481603.2022.2101727.

[6] Williams C., Falck F., Deligiannidis G., Holmes C., Doucet A., Syed S.: *A unified framework for U-net design and analysis*, [in:] Oh A., Naumann T., Globerson A., Saenko K., Hardt M., Levine S. (eds.), *NIPS '23: Proceedings of the 37th International Conference on Neural Information Processing Systems*, Curran Associates, Red Hook 2023, pp. 27745–27782.

[7] Xu Y., Wu L., Xie Z., Chen Z.: *Building extraction in very high resolution remote sensing imagery using deep learning and guided filters*. Remote Sensing, vol. 10(1), 2018, 144. https://doi.org/10.3390/rs10010144.

[8] Wang Y., Shang L., Liu Y., Sun Z., Lu Y., Shi X., Wang T., Liu Y.: *Precise building semantic segmentation in remote sensing images via MR-DeepLabv3+ network*. Scientific Reports, vol. 15(1), 2025, 40100. https://doi.org/10.1038/s41598-025-23917-9.

[9] He K., Gkioxari G., Dollár P., Girshick R.: *Mask R-CNN*, [in:] *2017 IEEE International Conference* on *Computer Vision (ICCVW 2017)*: *Venice*, *Italy 22–29 October 2017*, IEEE, Piscataway 2017, pp. 2961–2969. https://doi.org/10.1109/ICCV.2017.322.

[10] Kaya E.: *Assessment of building extraction and geometric corrections from UAV image using Mask R-CNN deep neural network algorithm*. European Journal of Remote Sensing, vol. 59(1), 2026, 2618374. https://doi.org/10.1080/22797254.2026.2618374.

[11] Han K., Wang Y., Chen H., Chen X., Guo J., Liu Z., Tang Y., Xiao A., Xu C., Xu Y., Yang Z., Zhang Y., Tao D.: *A survey on vision transformer*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45(1), 2023, pp. 87–110. https://doi.org/10.1109/TPAMI.2022.3152247.

[12] Gibril M.B.A., Al-Ruzouq R., Shanableh A., Jena R., Bolcek J., Shafri H.Z.M., Ghorbanzadeh O.: *Transformer-based semantic segmentation for large-scale building footprint extraction from very-high resolution satellite images*. Advances in Space Research, vol. 73(10), 2024, pp. 4937–4954. https://doi.org/10.1016/j.asr.2024.03.002.

[13] Zhang C., Liu L., Cui Y., Huang G., Lin W., Yang Y., Hu Y.: *A comprehensive survey on Segment Anything Model for vision and beyond*. arXiv preprint, arXiv:2305.08196, 2023. https://doi.org/10.48550/arXiv.2305.08196.

[14] Zhang J., Li Y., Yang X., Jiang R., Zhang L.: *RSAM-Seg: A SAM-based model with prior knowledge integration for remote sensing image semantic segmentation*. Remote Sensing, vol. 17(4), 2025, 590. https://doi.org/10.3390/rs17040590.

[15] Hafiz A.M., Bhat G.M.: *A survey on instance segmentation: State of the art*. International Journal of Multimedia Information Retrieval, vol. 9(3), 2020, pp. 171–189. https://doi.org/10.1007/s13735-020-00195-x.

[16] Mo Y., Wu Y., Yang X., Liu F., Liao Y.: *Review the state-of-the-art technologies of semantic segmentation based on deep learning*. Neurocomputing, vol. 493, 2022, pp. 626–646. https://doi.org/10.1016/j.neucom.2022.01.005.

[17] Kaur R., Singh S.: *A comprehensive review of object detection with deep learning*. Digital Signal Processing, vol. 132, 2023, 103812. https://doi.org/10.1016/j.dsp.2022.103812.

[18] Khanal B., Shrestha P., Amgain S., Khanal B., Bhattarai B., Linte C.A.: *Investigating the robustness of vision transformers against label noise in medical image classification*, [in:] *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2024): Orlando, Florida, USA, 15–19 July 2024*, IEEE, Piscataway 2024, pp. 1–6. https://doi.org/10.1109/EMBC53108.2024.10782929.

[19] Su H., Wei S., Yan M., Wang C., Shi J., Zhang X.: *Object detection and instance segmentation in remote sensing imagery based on precise Mask R-CNN*, [in:] *IEEE International Geoscience and Remote Sensing Symposium: Proceedings: July 28–August 2, 2019 Yokohama, Japan*, IEEE, Piscataway 2019, pp. 1454–1457. https://doi.org/10.1109/IGARSS.2019.8898573.

[20] Chen L.-C., Papandreou G., Schroff F., Adam H.: *Rethinking atrous convolution for semantic image segmentation*. arXiv preprint, arXiv:1706.05587, 2017. https://doi.org/10.48550/arXiv.1706.05587.

[21] Wang H., Yu F., Xie J., Wang H., Zheng H.: *Road extraction based on improved DeepLabv3 plus in remote sensing image*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLVIII-3/W2-2022, 2022, pp. 67–72. https://doi.org/10.5194/isprs-archives-XLVIII-3-W2-2022-67-2022.

[22] Chen L.-C., Zhu Y., Papandreou G., Schroff F., Adam H.: *Encoder-decoder with atrous separable convolution for semantic image segmentation*, [in:] Ferrari V., Hebert M., Sminchisescu C. (eds.), *Computer Vision – ECCV 2018: 15th European Conference: Munich, Germany, September 8–14, 2018: Proceedings, Part VII*, Springer, Cham 2018, pp. 833–851. https://doi.org/10.1007/978-3-030-01234-2_49.

[23] Felipe-García B., Hernández-López D., Lerma J.L.: *Analysis of the ground sample distance on large photogrammetric surveys*. Applied Geomatics, vol. 4(4), 2012, pp. 231–244. https://doi.org/10.1007/s12518-012-0084-2.

[24] Torgo L., Ribeiro R.: *Precision and recall for regression*, [in:] Gama J., Costa V.S., Jorge A.M., Brazdil P.B. (eds.), *Discovery Science: 12th International Conference, DS 2009, Porto, Portugal, October 3–5, 2009*, Springer, Berlin, Heidelberg 2009, pp. 332–346. https://doi.org/10.1007/978-3-642-04747-3_26.

[25] Foody G.M.: *Challenges in the real world use of classification accuracy metrics: From recall and precision to the Matthews correlation coefficient*. PLoS One, vol. 18(10), 2023, e0291908. https://doi.org/10.1371/journal.pone.0291908.

[26] Chicco D., Jurman G.: *The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation*. BMC Genomics, vol. 21(1), 2020, 6. https://doi.org/10.1186/s12864-019-6413-7.

[27] Zhang H., Zhang S.: *Focaler-IoU: More focused intersection over union loss*. arXiv preprint, arXiv:2401.10525, 2024. https://doi.org/10.48550/arXiv.2401.10525.

[28] QGIS Development Team: *QGIS geographic information system*. Open Source Geospatial Foundation Project, 2009. https://www.qgis.org.

[29] Sullivan J.: *Qgis-Plugin-Builder*. GitHub Repository, 2026. https://github.com/g-sherman/Qgis-Plugin-Builder [access: August 4, 2026].

[30] Blower J.D., Gemmell A.L., Griffiths G.H., Haines K., Santokhee A., Yang X.: *A web map service implementation for the visualization of multidimensional gridded environmental data*. Environmental Modelling & Software, vol. 47, 2013, pp. 218–224. https://doi.org/10.1016/j.envsoft.2013.04.002.

[31] Mandelbaum A., Weinshall D.: *Distance-based confidence score for neural network classifiers*. arXiv preprint, arXiv:1709.09844, 2017. https://doi.org/10.48550/arXiv.1709.09844.

[32] Bodla N., Singh B., Chellappa R., Davis L.S.: *Improving object detection with one line of code*, [in:] *2017 IEEE International Conference on Computer Vision (ICCVW 2017): Venice, Italy 22–29 October 2017*, IEEE, Piscataway 2017, pp. 5562–5570. https://doi.org/10.1109/ICCV.2017.593.

[33] Amari S.: *Backpropagation and stochastic gradient descent method*. Neurocomputing, vol. 5(4–5), 1993, pp. 185–196. https://doi.org/10.1016/0925-2312(93)90006-O.

[34] Zhou P., Xie X., Lin Z., Yan S.: *Towards understanding convergence and generalization of AdamW*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46(9), 2024, pp. 6486–6493. https://doi.org/10.1109/TPAMI.2024.3382294.

[35] Loshchilov I., Hutter F.: *Decoupled weight decay regularization*, [in:] *7th International Conference on Learning Representations: ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*, OpenReview.net, 2019. https://openreview.net/forum?id=Bkg6RiCqY7.

[36] Zhang G., Wang C., Xu B., Grosse R.B.: *Three mechanisms of weight decay regularization*. arXiv preprint, arXiv:1810.12281, 2018. https://doi.org/10.48550/arXiv.1810.12281.