ON THE ASSESSMENT OF INSTANCE SEGMENTATION FOR THE AUTOMATIC DETECTION OF SPECIFIC CONSTRUCTIONS FROM VERY HIGH RESOLUTION AIRBORNE IMAGERY

P. Achanccaray[1,*], M. Gerke[1], L. Wesche[2], S. Hoyer[2], K. Thiele[2], U. Knufinke[3], C. Krafczyk[3]

[1] Institute of Geodesy and Photogrammetry, Technical University of Braunschweig,
Braunschweig, Germany - (p.diaz, m.gerke)@tu-braunschweig.de
[2] Institute of Preservation of Buildings and Structure, Technical University of Braunschweig,
Braunschweig, Germany - (l.wesche, s.hoyer, k.thiele)@bauwerkserhaltung.tu-braunschweig.de
[3] Lower Saxony State Office for the Preservation of Monuments,
Hannover, Germany - (ulrich.knufinke, christina.krafczyk)@nld.niedersachsen.de

KEY WORDS: Deep Learning, Instance Segmentation, Remote Sensing, Building Detection, Building Heritage, Aerial Imagery.

ABSTRACT:

During the High Modernism period spanning from approximately 1914 to 1970, the manufacturing of steel-constructed system halls witnessed a significant surge to accommodate the growing demand across various sectors such as industry, commerce, and agriculture. Surprisingly, these specific types of buildings have been largely overlooked in the realm of construction history research, resulting in a dearth of knowledge regarding their construction methods, distribution patterns, and contextual significance for assessing their historical value. This study aims to address this gap by exploring the potential of instance segmentation methods for the automated detection of system halls using high-resolution aerial imagery. To achieve this objective, state-of-the-art deep learning models are evaluated in terms of their ability to localize and delineate system halls accurately. Our experiments reveal that Mask R-CNN yields the most accurate results both quantitatively and qualitatively, closely followed by Cascade Mask R-CNN. However, it is important to note that multi-scale methods may introduce false positives since system halls possess distinct geometric dimensions that necessitate careful consideration during the detection process.

1. INTRODUCTION

Between 1914 and 1970, the High Modernism period saw the widespread production of various steel system halls to attend to the growing need for new spatial solutions driven by the industrial production and logistics requirements of medium-sized businesses. Regrettably, despite their prevalence, the construction history of system halls has been disregarded, leading to inadequate knowledge of the diversity of construction types, their distribution, and site-specific contexts. Consequently, it is difficult to determine the feasibility and suitability of listing them as historical monuments or recognizing them as valuable and sustainable structures. This work aims to contribute to the research on system halls by developing methods for their automatic detection. The precise location of these buildings will allow further study and their evaluation as objects in the history of building construction and as potential monuments.

Automatic building detection can be performed using machine learning (ML) and deep learning (DL) algorithms. The primary distinction between ML and DL is the feature extraction stage required by the former. This stage typically involves a specialist who identifies the optimal set of features based on their expertise (e.g., geometric dimensions, texture, appearance, color, and shape). In contrast, DL algorithms enable end-to-end learning during training, where the algorithm learns task-specific features automatically. In this context, we recognized three main methods (see Figure 1) for building detection based on DL: semantic segmentation, object detection, and instance segmentation. The semantic segmentation output is a mask in which each pixel is associated with a class label (e.g., system hall and background). Object detection provides boxes enclosing each object (building) and its corresponding class. Instance segmentation produces a more structured output with boxes enclosing each instance (individual buildings) of each class and a mask with a class label for each pixel.

Semantic segmentation needs a post-processing (e.g., identification of connected components) stage to isolate individual instances and to obtain their precise localization (i.e., bounding box coordinates). On the other hand, object detection lacks a proper delineation of each building footprint, so it is necessary to apply another method to delineate each building inside the generated bounding box. In this context, instance segmentation supplies better management of individual instances for the later extraction of measurements (e.g., geometric dimensions and roof shape) from each building. These measurements are employed in the categorization and evaluation of each system hall.

There are four main categories of approaches for instance segmentation: classification of mask proposals (Hariharan et al., 2014), detection plus segmentation (Zhou et al., 2019), segmentation plus clustering (Kirillov et al., 2017), and dense sliding windows (Pinheiro et al., 2015). In the first approach, the method involves proposing mask candidates (Arbeláez et al., 2014) and extracting features from these masks. Afterwards, these masks are classified and refined to enhance the accuracy of object boundaries. The second approach modifies architectures similar to two-stage detectors to generate object masks. For example, Mask R-CNN (Zhou et al., 2019) extends Faster R-CNN (Girshick, 2015) by incorporating an object mask prediction branch along with the object bounding box recogni-

* Corresponding author

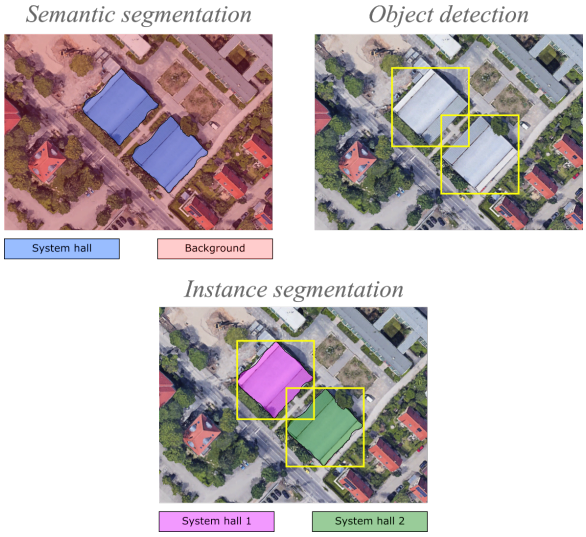

Figure 1. Exemplification of results obtained different approaches for automatic building detection from aerial imagery.

tion branch. The third approach combines semantic segmentation with clustering methods to distinguish between object instances and backgrounds. In (Kirillov et al., 2017), two separate branches are used to produce per-pixel class and edge scores. The edge scores are employed to extract super-pixels, which are then merged based on the class scores to generate the object instances. Lastly, the fourth approach applies a dense sliding window technique across the entire image, where each window passes through a model equipped with segmentation and object detection heads. In (Pinheiro et al., 2015), these heads are parallel branches that predict a segmentation mask for the object located at the center and an object score to determine the presence of an object within the window.

In our previous work (Achanccaray et al., 2023), we extensively assessed semantic segmentation and object detection methods for system hall detection from very high-resolution airborne imagery, concluding that segmentation methods are more suitable for this task. In this work, we explore instance segmentation methods as a better solution for the automatic detection of system halls. Thus, we aim to answer the following research questions:

- Is it possible to directly detect specific building types in an end-to-end learning manner without a post-processing step?
- Is instance segmentation more suitable than other methods for the detection of specific building types?

To this end, we use state-of-the-art instance segmentation methods to assess their suitability for automatic system hall detection. The evaluation of these methods is performed quantitatively and qualitatively.

The remaining parts of this work are organized as follows: Section 2 presents the study area and the dataset built to train and assess all methods, Section 3 explains each step of the methodology employed for automatic detection of system halls and provides a brief description of each method, Section 4, describes the experimental protocol followed in our experiments and the results obtained by each method in terms of accuracy metrics and visual predictions, and Section 5 summarizes our findings based on the performed experiments and outlines the next steps in our research to further improve our results

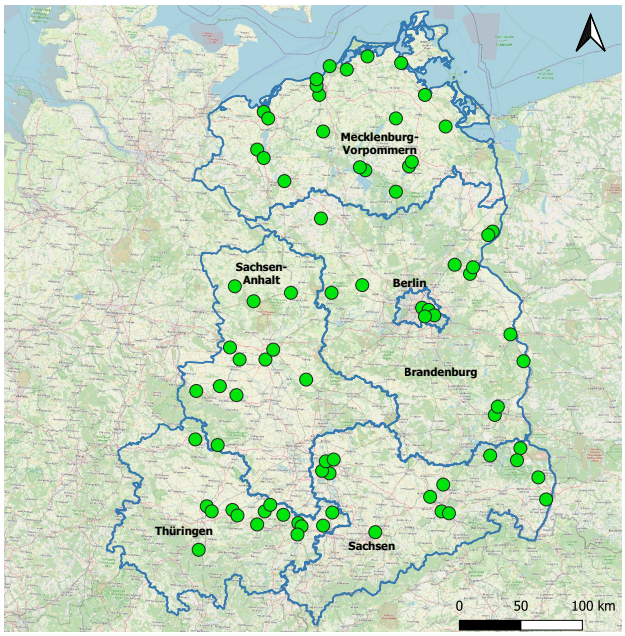

Figure 2. Distribution of system hall locations found by visual inspection.

2. DATASET

Our dataset was built based on the locations of system halls found by visual inspection using aerial photos and 3D views from Google Earth. This inspection was performed employing the information obtained from different system hall manufacturers from the former German Democratic Republic (GDR). Then, Digital Orthophotos (DOP) and Normalized Digital Surface Models ($nDSM$) were collected to cover all these locations (green dots in Figure 2). A total of 110 image tiles were used from different states in Germany: Berlin (7), Brandenburg (16), Mecklenburg-Vorpommern (29), Sachsen (20), Sachsen-Anhalt (15), and Thüringen (23). These tiles have sizes of $5\text{K} \times 5\text{K}$ or $10\text{K} \times 10\text{K}$ pixels, with 20 cm spatial resolution. $DOPs$ contain spectral information in four channels: Red, Green, Blue, and Infrared. $nDSM$ was created by the subtraction between Digital Surface Models (DSM) and Digital Elevation Models (DEM). Both spectral and height information was acquired from each state's geoportals[1234].

The annotations (labels) to train the detection algorithms for instance segmentation were created manually using the Quantum GIS (QGIS)[5] software. First, a vector layer was created to store the polygons delineating each building. Then, each vector layer was rasterized in TIF format. Finally, the segmentation mask and bounding box coordinates were obtained from the delineated polygons and the coordinates of their contours. Our dataset includes four different types of system halls (see Figure 3): *KT 60 L*, *Ruhland*, *GT 60 L*, and *24×42*. Figure 3 shows each system hall type illustration (first row) and 20 cm spatial resolution DOP (second and third rows) covering each type location. The total number of instances is 188 distributed in the following way: *KT 60 L* (81), *Ruhland* (74), *GT 60 L* (10), and *24×42* (23). As our main objective is to find where a system hall is

[1] https://www.geodaten.sachsen.de
[2] https://www.lvermgeo.sachsen-anhalt.de/
[3] https://www.geoportal-th.de/
[4] https://geobasis-bb.de/lgb/de/
[5] Available at: https://qgis.org/en/site/index.html

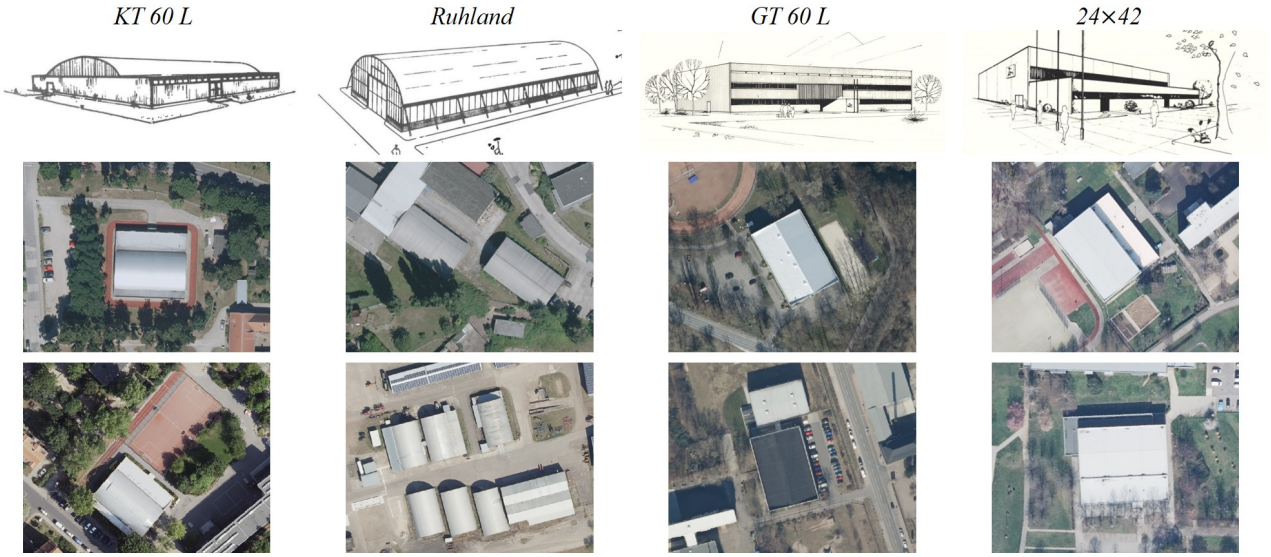

Figure 3. System hall types available in our dataset. From left to right: *KT 60 L*, *Ruhland*, *GT 60 L*, and *24×42*. Sketches (first row) and their respective 20 cm DOP (second and third rows).

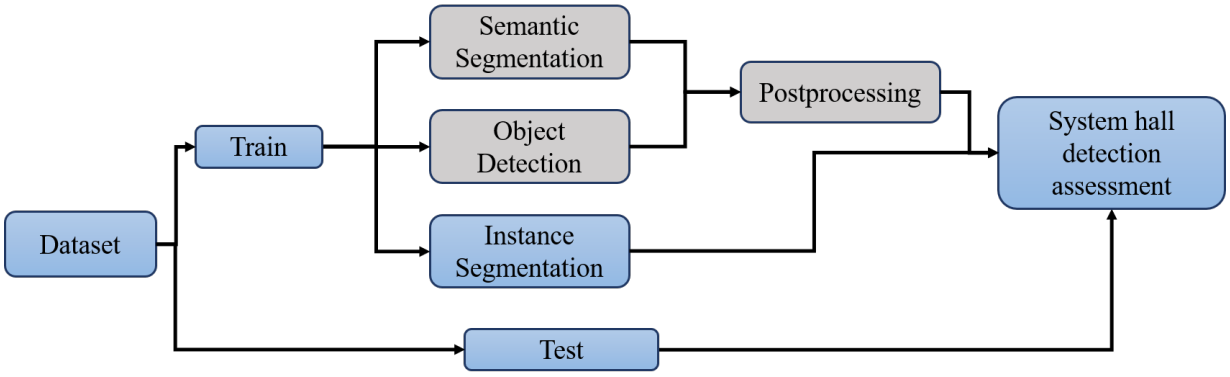

Figure 4. Methodology followed to assess instance segmentation models (blue boxes) for the automatic detection of system halls, in comparison with our previous work where semantic segmentation and object detection methods were explored (grey boxes).

located independently of its type, we merged all classes into a single class. For that reason, our dataset comprises two classes: *system hall* and *background*.

3. METHODOLOGY

Our methodology is presented in Figure 4, where instance segmentation methods are used for the automatic detection of system halls. In contrast with semantic segmentation and object detection approaches, instance segmentation provides us directly with the precise localization and mask of each building, not needing post-processing of their outcome. From the entire dataset, two mutually exclusive sets are created: train and test. The train set is split internally into two sets: train and validation (not shown in Figure 4), where the train set is used to update the parameters model and the validation set to select the best set of these parameters. Once the training has finished, the test set is used to produce the model's prediction, which is assessed quantitatively and qualitatively to determine which model is the most suitable for our application.

For instance segmentation, we selected the following DL-based architectures for being the current state-of-the-art in the COCO[6] (Lin et al., 2014) dataset: Mask R-CNN, Cascade R-CNN (Cai and Vasconcelos, 2018), Mask Scoring R-CNN (Huang et al., 2019), SOLOv2 (Wang et al., 2020b), and RTMDet (Lyu et al., 2022).

Mask R-CNN: Mask R-CNN extends Faster R-CNN by including a branch for predicting an object mask parallel to the bounding box recognition branch. Thus, the first stage of Mask R-CNN is similar to Faster R-CNN: region proposal. Then, in the second stage, Mask R-CNN predicts a binary mask for each Region of Interest (RoI), a class, and box offsets.

Cascade R-CNN: Cascade R-CNN was proposed to overcome the detection performance decreasing as a consequence of using high Intersection over Union (IoU) during training. Cascade R-CNN is composed of a series of detectors trained using progressively higher IoU thresholds, enabling them to become increasingly discerning when it comes to eliminating nearby false positives. The detectors undergo incremental training stages, taking advantage of the insight that the output of one detector serves as a valuable training distribution for the subsequent detector of superior quality.

Mask Scoring R-CNN: Mask Scoring R-CNN deals with the lack of correlation between mask quality and its classification

[6] https://cocodataset.org/

		bounding box			*mask*		
Model	Backbone	$AP^{0.5}$	$AP^{0.75}$	$AP^{0.5:0.95}$	$AP^{0.5}$	$AP^{0.75}$	$AP^{0.5:0.95}$
Mask R-CNN	ResNet 50	**84.6**	**76.1**	**66.5**	84.6	70.6	**67.8**
Cascade Mask R-CNN	ResNet 50	76.8	72.4	65.9	77.3	**72.4**	65.7
Mask Scoring R-CNN	ResNet 50	83.8	71.1	61.4	82.8	70.0	63.9
SOLOv2	ResNet 50	-	-	-	**87.1**	68.7	65.3
RTMDet *tiny*	CSPDarknet	65.3	53.4	44.6	68.3	57.2	52.0
RTMDet *small*	CSPDarknet	55.7	43.1	34.1	57.4	41.6	41.4

Table 1. AP values at different IoU thresholds for bounding box and mask obtained by all selected methods. The highest values per threshold are highlighted in bold.

score. Within Mask Scoring R-CNN, a network block is dedicated to acquiring knowledge about the quality of the predicted instance masks. This network block combines the instance feature with the corresponding predicted mask, allowing it to estimate the mask IoU through regression. The mask scoring strategy addresses any disparities between mask quality and mask score, enhancing the performance of instance segmentation by giving higher priority to more precise mask predictions.

SOLOv2: SOLOv2 is an improved version of SOLO (Segmenting Objects by LOcations) (Wang et al., 2020a). SOLOv2 derives its strength from a proficient and comprehensive instance mask representation scheme that actively segments each instance present in the image, bypassing the need for bounding box detection. More precisely, the process of generating object masks in SOLOv2 is divided into two parts: mask kernel prediction and mask feature learning. The former generates convolution kernels, while the latter generates the feature maps that are subsequently convolved with these kernels. For this reason, SOLOv2 generates a mask per instance without direct prediction of its bounding box.

RTMDet: RTMDet uses modified CSPDarknet blocks (Bochkovskiy et al., 2020) with large kernel depth-wise convolution layers as the backbone. The *neck* of the model is composed of multilevel features extracted from the backbone and fused by a series of convolutions. Finally, the detection heads have shared convolution weights and are used to predict the classification and regression results for (rotated) bounding box detection. In our experiments, we only used the *tiny* and *small* versions of RTMDet due to memory limitations.

4. RESULTS

4.1 Experimental protocol

The 110 image tiles acquired in the dataset were split into three mutually exclusive sets: train (44), validation (22), and test (44). These splits were created in a stratified manner considering the German states of each image tile to ensure that each set has images from all states. This is important as image tiles from different states are slightly different in terms of appearance due to different pre-processing approaches, especially for $nDSM$, which depending on the state, can be image- or LiDAR-based with different point densities and grid sizes. Then, from each image tile, patches of 1500×1500 were extracted from each building. A total of five patches were extracted from each building by varying the relative position of each building in the patch: centered, to the north, to the south, to the west, and to the east. This was performed to increase the number of patches and the variability of samples in our dataset. Finally, patches with an overlapping higher than 50% were discarded to avoid redundant information.

All DL instance segmentation methods used ResNet 50 (He et al., 2016) as backbone pre-trained with ImageNet (Deng et al., 2009), with the exception of RTMDet, which uses CSPDarknet. We used each model's pre-trained weights with the COCO dataset and fine-tuned the models for a maximum of 50 epochs. All model parameters are the ones recommended by the authors. The whole methodology was implemented in Python language using PyTorch and the MMDetection[7] framework. To increase each model's robustness, random flips were applied as data augmentation during training. Only this transformation was considered to avoid modifying the scale and appearance of the buildings because system halls have specific geometric dimensions and roof materials. As the input to these models is an image with three channels, we created false color RGB compositions to train each model. The false color RGB composition has the following configuration: Infrared (R), Gray (G), and $nDSM$ (B), where Gray is the arithmetic mean of the Red, Green, and Blue channels from the DOP, with equal weights for each channel.

4.2 Quantitative results

Table 1 summarizes our results in terms of Average Precision (AP) at different IoU thresholds: 0.5, 0.75, and 0.5:0.95, for bounding box and mask obtained by all instance segmentation methods. The highest values per threshold are highlighted in bold. The bounding box metrics provide information regarding how accurate is the localization of each building, while mask metrics about their delineation at the pixel level.

Mask R-CNN achieved the highest AP for almost all thresholds, while the lighter versions of RTMDet (*tiny* and *small*) got the worst results. Note that there are no bounding box results for SOLOv2 as this algorithm directly produces a mask per instance. A good trade-off between bounding box and mask metrics is provided by Mask R-CNN and Cascade Mask R-CNN, which obtained the best results for $\mathrm{IoU}^{0.5:0.95}$. Mask Scoring R-CNN and SOLOv2 obtained high mask AP for $\mathrm{IoU}^{0.5}$: 82.8 and 87.1, respectively; however, it decreased substantially for $\mathrm{IoU}^{0.5:0.95}$, indicating that these models are not as robust as Mask R-CNN.

[7] Available at: https://github.com/open-mmlab/mmdetection

Figure 5. Snips of images from the test set with their corresponding predictions and scores (black boxes) for all selected methods. From left to right: samples acquired from different locations. Note that all models were trained on false color RGB compositions but, just for visualization purposes, the true color RGB compositions with the predictions are shown.

4.3 Qualitative results

Figure 5 shows snips of images from the test set with their corresponding predictions and scores (i.e., how confident the model is about the prediction) generated by each method (each row).

We can observe in the first column (Figure 5) that occluded (e.g., by trees) and too-close instances (i.e., a building next to the other) are challenging for almost all methods. For instance, Mask Scoring R-CNN and RTMDet, both versions, missed one or two buildings (RTMDet *tiny*), while SOLOv2 produced a mask delineating both buildings but as a single instance as the buildings are adjacent. In addition, only Mask R-CNN obtained predictions with high scores (73.9 and 98.0), while other methods were uncertain about their results (e.g., SOLOv2: 56.8 and RTMDet *small*: 51.1).

In the second column of Figure 5, we can see the presence of some false positives and negatives. Cascade Mask R-CNN and RTMDet *tiny* missed one building, while RTMDet *small* generated two false positives, which correspond to a large building with a similar roof. In general terms, almost all methods successfully delineated each building, with the exception of RTMDet *small*, which had some problems with shadows (Figure 5, third column) resulting in a concave mask.

Note that in the last column of Figure 5, all methods detected the same false positive. From the DOP, this building has a similar roof type to the *GT 60 L* type (see Figure 3). To verify if it is a new finding or just a false positive, we collected 3D views of that location from Google Earth. Figure 6 presents these 3D views and their corresponding DOP for a *GT 60 L* type (left) and the aforementioned false positive (right).

We can see that both buildings look really similar; however, the false positive corresponds to a greenhouse with a similar outer structure but different geometric dimensions (height, width, and length). Even if we did not apply any data augmentation technique to simulate different scales, many instance segmentation methods do it to be robust against objects at multiple scales. This is performed by resizing the input image to different predefined input sizes depending on the method. This is a particular characteristic of state-of-the-art instance segmentation methods which is not desired for our specific application as the buildings we are looking for possess exact geometric dimensions. Thus, instance segmentation is suitable for our application based on the proper localization and delineation of each building; however, it is necessary to consider the robustness of the methods against different scales, which can generate false positives as described before.

5. CONCLUSIONS

In this work, we explored the suitability of instance segmentation models for the automatic detection of system halls from the high-modernism period using aerial imagery. For this purpose, state-of-the-art DL-based algorithms were selected and assessed quantitatively (in terms of average precision at different IoU thresholds) and qualitatively (based on a visual inspection of the generated predictions). From our experiments, we concluded that instance segmentation methods are able to generate accurate localization and delineation of system halls, especially Mask R-CNN, which achieved the highest metric values. It is worth mentioning that the multi-scale characteristic

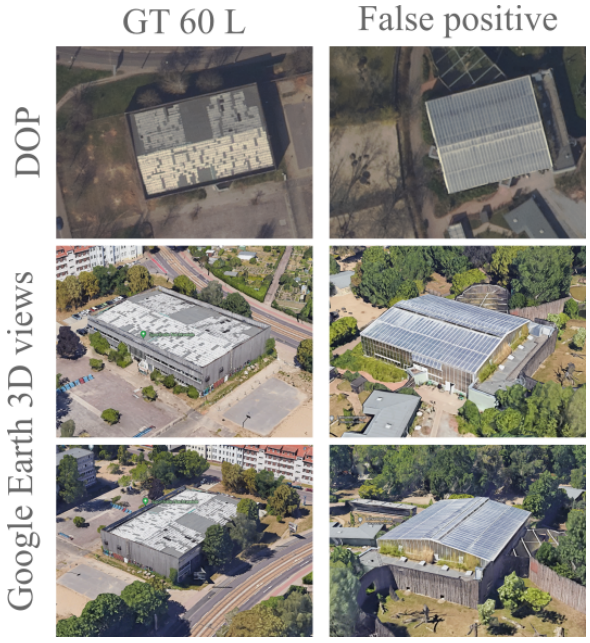

Figure 6. DOP and Google Earth 3D views of a *GT 60 L* and a false positive detected by all methods.

of some methods can lead to the generation of false positives. This is critical for our application as system halls are parametrized buildings that were produced with specific geometric dimensions and scales. This can be overcome by fixing the scales to train each method, or post-processing to discard those false positives.

In the following steps, we will perform the automatic detection of system hall types. This is a multi-class problem that we have not explored yet as there are some system hall types with just a few instances, resulting in a highly imbalanced dataset. Moreover, based on the information collected from manufacturers, there are more than 80 system hall types in Germany, of which some types might not continue to exist nowadays. To overcome the lack of samples for some types, we will use synthetic data generated manually or automatically by generative models such as Generative Adversarial Networks (GANs) or Diffusion models.

REFERENCES

Achanccaray, P., Gerke, M., Wesche, L., Hoyer, S., Thiele, K., Knufinke, U., Krafczyk, C., 2023. Automatic Detection of Specific Constructions on a Large Scale Using Deep Learning in Very High Resolution Airborne Imagery: The Case of Steel Construction System Halls of the High Modernism Period. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 1–21. doi.org/10.1007/s41064-023-00237-z.

Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J., 2014. Multiscale Combinatorial Grouping. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 328–335.

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M., 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Cai, Z., Vasconcelos, N., 2018. Cascade R-CNN: Delving Into High Quality Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Girshick, R., 2015. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Hariharan, B., Arbeláez, P., Girshick, R., Malik, J., 2014. Simultaneous Detection and Segmentation. D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (eds), *Computer Vision – ECCV 2014*, Springer International Publishing, 297–312.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X., 2019. Mask Scoring R-CNN. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., Rother, C., 2017. InstanceCut: From Edges to Instances with MultiCut. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 7322–7331.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Doll'a r, P., Zitnick, C. L., 2014. Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312. http://arxiv.org/abs/1405.0312.

Lyu, C., Zhang, W., Huang, H., Zhou, Y., Wang, Y., Liu, Y., Zhang, S., Chen, K., 2022. RTMDet: An Empirical Study of Designing Real-Time Object Detectors.

Pinheiro, P. O., Collobert, R., Dollár, P., 2015. Learning to Segment Object Candidates. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, MIT Press, Cambridge, MA, USA, 1990–1998.

Wang, X., Kong, T., Shen, C., Jiang, Y., Li, L., 2020a. Solo: Segmenting objects by locations. A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (eds), *Computer Vision – ECCV 2020*, Springer International Publishing, Cham, 649–665.

Wang, X., Zhang, R., Kong, T., Li, L., Shen, C., 2020b. SOLOv2: Dynamic and Fast Instance Segmentation. H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (eds), *Advances in Neural Information Processing Systems*, 33, Curran Associates, Inc., 17721–17732.

Zhou, K., Chen, Y., Smal, I., Lindenbergh, R., 2019. Building segmentation from airborne VHR images using Mask R-CNN. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42(2/W13).