www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

# TRAINING DATA GENERATION FOR CONSTRUCTION INSTANCE SEGMENTATION FROM TARGET DOMAINS

SUBMITTED: December 2025
PUBLISHED: May 2026
EDITOR: Frédéric Bosché
DOI: 10.36680/j.itcon.2026.028

***Xuzhong Yan, Lecturer***
***Department of Construction Management, School of Management, Zhejiang University of Technology, Hangzhou 310023, China***
***https://orcid.org/0000-0002-1675-2368***
***xzyan@zjut.edu.cn***

***Zeli Wang, Lecturer***
***School of Business, East China University of Science and Technology, Shanghai 200237, China***
***https://orcid.org/0000-0002-5613-6148***
***wang.zeli@ecust.edu.cn***

***SUMMARY***: *This study addresses a persistent challenge in computer vision for construction monitoring: deep learning models trained on source-domain data often perform poorly when deployed in new target domains due to distribution shifts and limited annotations. To mitigate these issues, the research introduces TDG-CIS, a clustering-initialized semi-supervised framework designed to generate high-quality instance segmentation training data directly from unlabeled target-domain images. TDG-CIS operates in two stages. First, it employs a clustering-based mask generation strategy that uses a transformer feature backbone to extract patch-level representations and derive initial instance masks without human supervision. These masks serve as a reliable starting point for semi-supervised learning. Second, a semi-supervised instance segmentation model iteratively refines these masks and converts raw images into usable training samples. This iterative pipeline allows the model to progressively improve segmentation quality while adapting to the visual characteristics of diverse construction environments. The framework was validated on a large dataset of 50,000 images spanning more than 70 construction-related domains. Experimental results show that TDG-CIS achieves a 77.9% data utilization rate, along with 87.5% mAP and 81.1% mAR in segmentation quality. When used to scale training data for downstream instance segmentation models, TDG-CIS yields substantial performance gains: baseline models trained on automatically generated data outperform those trained on manually labeled datasets, improving mAP from 92.9% to 94.3% and mAR from 86.7% to 88.6%. Ablation studies further demonstrate that the semi-supervised refinement mechanism is key to boosting both data utilization and segmentation accuracy. Overall, the study offers a novel approach that eliminates dependence on source-domain supervision and provides a scalable pathway for producing target-domain training datasets for instance segmentation in intelligent construction applications.*

***KEYWORDS***: *training data generation, clustering, semi-supervised, construction instance segmentation.*

***REFERENCE***: *Yan, X., & Wang, Z. (2026). Training data generation for construction instance segmentation from target domains. Journal of Information Technology in Construction (ITcon), 31, 632-650. https://doi.org/10.36680/j.itcon.2026.028*

# 1. INTRODUCTION

Even being trained on large-scale datasets, deep learning-based computer vision (DLBCV) models remain prone to performance degradation in target domains that differ from their source training domain (Hong et al., 2023, Kim et al., 2018), owing to their sensitivity to cross-domain data distribution shifts (Chern et al., 2023). This issue is especially pronounced in construction, where each project's unique and temporary layout creates substantial shifts that undermine model robustness and generalization (Zha et al., 2025).

A straightforward solution is to manually label training data for new target domains. However, this process is error-prone, labor-intensive and time-consuming, and cannot keep pace with increasingly complex DLBCV models, creating a widening gap between dataset availability and model scalability (Mondal et al., 2025).

Alternative methodologies have been explored to enhance the performance of DLBCV models in new target domains while reducing reliance on manual labeling, including non-fully-supervised paradigms (e.g., unsupervised (Weng et al., 2023), semi-supervised (Kim et al., 2025), and self-supervised (Chern et al., 2023) learning), domain adaptation (Huang et al., 2024), n-shot learning (Yong et al., 2023), and synthetic data generation (Zhang et al., 2024). However, two common issues remain in the existing related works. First, although prior works have achieved encouraging improvements in model performance in target domains, they still require source training domain to reach their full potential, making data annotation unavoidable (Gupta et al., 2024, Kim et al., 2025). Second, previous studies focus on object detection or semantic segmentation, which provide coarse localization or category-level labeling, while limited attention has been given to instance segmentation, which requires precise instance-level delineation.

To address these gaps, this study aims to enable the direct generation of instance segmentation training data in target domains without relying on source training domain, thereby simultaneously alleviating annotation cost and improving cross-domain generalization. Accordingly, this study proposes a clustering-initialized semi-supervised training data generation method for construction instance segmentation (TDG-CIS) from target domains. TDG-CIS converts raw visual data directly into training data with instance-level contours and category annotations, which can be utilized for a variety of computer vision tasks in construction, such as object detection, instance segmentation, and multi-object tracking. Experimental results demonstrate that TDG-CIS achieves a 77.9% data utilization rate, 87.5% mAP, and 81.1% mAR on a large-scale public dataset covering over 70 construction domains. In addition, models trained on TDG-CIS-generated data from the same raw visual data pool achieve comparable performance to those trained on manually labeled data (mAP 86.8% vs. 92.9%, mAR 79.6% vs. 86.7%). Expanding the raw visual data pool with additional TDG-CIS-generated samples further enhances performance (mAP 94.3%, mAR 88.6%), surpassing that of models trained solely on manually labeled data. The effectiveness of the proposed method was further supported by an ablation study.

The contributions of this work are twofold: (1) we propose a clustering-initialized semi-supervised training data generation method for construction instance segmentation (TDG-CIS), which enables high-quality instance-level labeling and supports model training and inference in target domains without relying on source training domains; and (2) TDG-CIS achieves superior data utilization and model performance on a large-scale benchmark encompassing over 70 construction domains.

# 2. RELATED WORKS

## 2.1 Instance segmentation in construction vision tasks

Intelligent construction marks a shift toward integrating digital and intelligent technologies to enhance efficiency, safety, quality, and sustainability (Wang et al., 2025). Deep learning-based computer vision (DLBCV) plays a central role by enabling machines to interpret visual data from construction sites (Xu et al., 2020). Its importance is supported by advances in AI and deep learning (LeCun et al., 2015), improved computing hardware, and widespread deployment of cameras, drones, and robots in construction environments (Bock, 2015). Through visual analysis, computer vision enables real-time monitoring and decision-making, such as safety inspection (Kim et al., 2023a), quality assessment (Liang et al., 2023), and schedule evaluation (Yan et al., 2025).

Within this context, Instance segmentation has become a crucial computer vision task, providing pixel-level localization and differentiation of objects (Minaee et al., 2022). This capability is essential for interpreting complex

construction scenes, supporting hazard detection, progress monitoring, and autonomous decision-making, such as worker behavior analysis (Mei et al., 2024), machinery logistics (Yan et al., 2025), material management (Sun et al., 2025), and improved perception in robotic systems. However, the generalization ability of instance segmentation in construction has long been constrained by two major challenges. First, the scarcity of source domain training data. High-precision instance segmentation requires large-scale, high-quality source domain training data, yet public construction-specific training data (An et al., 2021, Yan et al., 2023) remain limited in both quantity and object categories compared with general large-scale benchmarks (Deng et al., 2024, Gupta et al., 2019), which constrain the ability of models to learn diverse visual features and generalize effectively from source domains. Second, data distribution shifts between source and target domains. This challenge is reflected in the vulnerability of DLBCV models to performance drops under cross-domain shifts. In construction, the uniqueness and temporality of projects cause substantial distribution variations that weaken robustness and generalization (Zha et al., 2025). Even models trained on large datasets like COCO often underperform across diverse domains (Hong et al., 2023).

These challenges have motivated this research to enhance instance segmentation generalization in target construction domains, particularly when source domain data is limited or unavailable.

## 2.2 Learning under data scarcity and distribution shifts

To mitigate manual labeling and improve DLBCV performance in new domains, two main strategies are used: non-fully-supervised learning, which exploits unlabeled or partially labeled data, and transfer learning, which adapts source knowledge to target domains. In practice, they are often combined to boost generalization.

Non-fully-supervised learning encompasses unsupervised, weakly supervised, semi-supervised, and self-supervised approaches. Unsupervised methods leverage inherent data structures, e.g., for structural defect inspection (Midwinter et al., 2023) or cross-material crack detection via domain adaptation (Weng et al., 2023). Semi-supervised learning combines a small set of labeled data with abundant unlabeled data to improve performance using techniques like pseudo-labeling and self-training (Amini et al., 2025). Applications include semantic segmentation (Hong et al., 2023) and object detection (Kim et al., 2025). For example, a recent study proposed a semi-supervised self-training framework for generating training data for object detection in construction (Kim et al., 2025). The method integrates optical flow estimation and iterative pseudo-label propagation to generate target-domain training samples. Self-supervised learning creates supervisory signals via pretext tasks, allowing models to leverage large unlabeled datasets (Ghelmani & Hammad, 2023). For example, Chern et al. (2023) proposed SESC-CAM, a fusion architecture combining weakly supervised and self-supervised learning for pseudo-label generation in semantic segmentation, demonstrating the effectiveness of this paradigm under limited annotation conditions.

As specialized transfer learning methods, domain adaptation and n-shot learning address data scarcity and distribution shifts in construction tasks. Examples include synthetic-to-real adaptation for rail inspection (Huang et al., 2024), domain-adaptive object detection for site monitoring (Kim et al., 2024a), and prompt optimization for defect detection (Yong et al., 2023).

Although the above methodologies mitigate data scarcity and distribution shifts and reduce reliance on extensive labeling, generating training data for instance segmentation from the target domain remains unaddressed, yet it is essential to capture pixel-level visual features and achieve optimal performance for DLBCV in practical applications.

## 2.3 Interactive segmentation tools and synthetic data generation

Both interactive segmentation tools (ISTs) and synthetic data generation aim to reduce manual labeling. ISTs use minimal human input, e.g., clicks or scribbles, to assist annotation. Examples include SimpleClick, a Vision Transformer-based IST (Liu et al., 2023) and InterFormer, which reduces latency (Huang et al., 2023). Nonetheless, ISTs still rely on human correction, limiting efficiency for large-scale construction datasets.

In contrast, synthetic data generation aims to automatically create labeled training data by simulating realistic construction objects and scenarios (Barrera-Animas et al., 2023, Kim et al., 2024b). For instance, Zhang et al. (2024) proposed a generative model to synthesize pavement crack images. Kim et al. (2023b) demonstrated a hybrid deep learning method leveraging both synthetic and real construction images. While synthetic data

generation methods can facilitate model training, they often rely on 3D models or source domain data, and may face challenges such as domain gaps, limited visual diversity, and unrealistic scene compositions, which constrains their utility in directly producing labeled training data from the target domain.

## 2.4 Research gap

This review highlights two critical research challenges in construction instance segmentation: limited target-domain training data and severe cross-domain distribution shifts. Although existing methodologies, including non-fully-supervised learning, transfer learning, interactive segmentation, and synthetic data generation, have shown promising progress, they remain insufficient for target-domain training data generation in practical construction scenarios. Specifically, they still require source training domain to reach their full potential, making data annotation unavoidable. In addition, prior studies predominantly focus on object detection or semantic segmentation, which provide coarse localization or category-level labeling, while the automatic generation of instance-level annotations remains largely unexplored. This limitation is particularly critical in construction monitoring, where precise instance-level delineation is essential for downstream tasks such as object tracking and activity analysis. Therefore, an effective method that can directly generate high-quality instance segmentation training data from unlabeled target domains without relying on source training domain remains a key research gap.

# 3. METHODOLOGY

## 3.1 Overall framework

To address the identified research gap, this paper presents a clustering-initialized semi-supervised training data generation method for construction instance segmentation (TDG-CIS) in unlabeled target domains without relying on source training domains. The overall framework is illustrated in Figure 1. The framework introduces a clustering-based mask generation method using a transformer-based feature backbone to extract patch tokens and derive instance masks. A semi-supervised model is then proposed to iteratively transform raw images into training data based on these masks. The proposed framework is a unified pipeline specifically developed to address the research problem of training data generation in unlabeled target domains for construction instance segmentation. Its significance lies in providing a unified pipeline specifically developed to address the longstanding challenge of target-domain training data generation for construction instance segmentation in construction monitoring. In this way, the framework enables progressive generation of instance-level annotations directly from target-domain images and provides a scalable pathway for deployment in real-world construction scenarios.

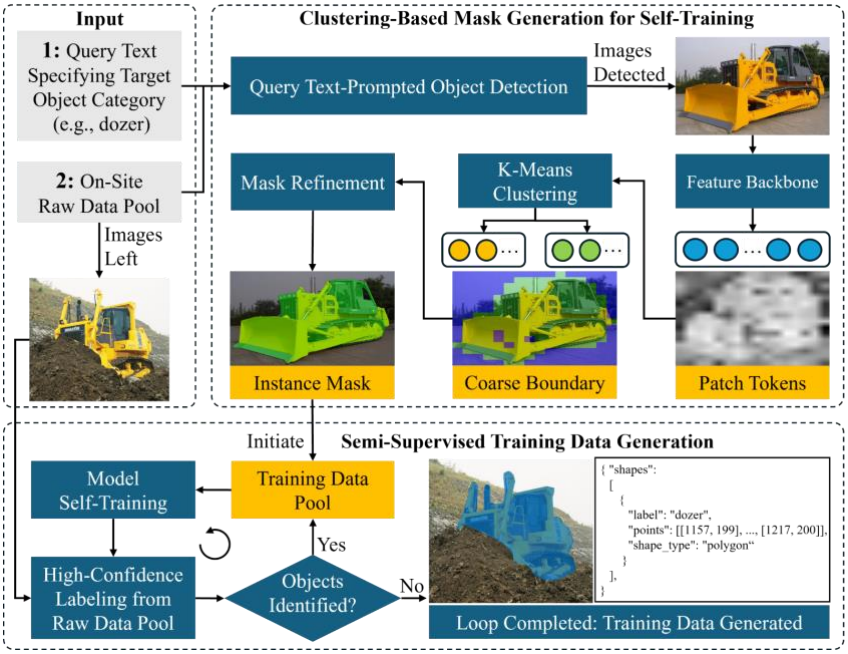


*Figure 1: Overall methodology framework.*

## 3.2 Clustering-based mask generation

The process of clustering-based mask generation method is visualized in Figure 2. The raw images are first deduplicated and then filtered using Grounding DINO with query text prompts corresponding to each target category (Liu et al., 2024). Each filtered object is then cropped as an individual image. On this set of cropped images, we generate instance masks for the single object in each image in an unsupervised manner by leveraging rich visual representations from a vision transformer (ViT)-based architecture (Oquab et al., 2024) and the segmentation capabilities of SAM (Kirillov et al., 2023).

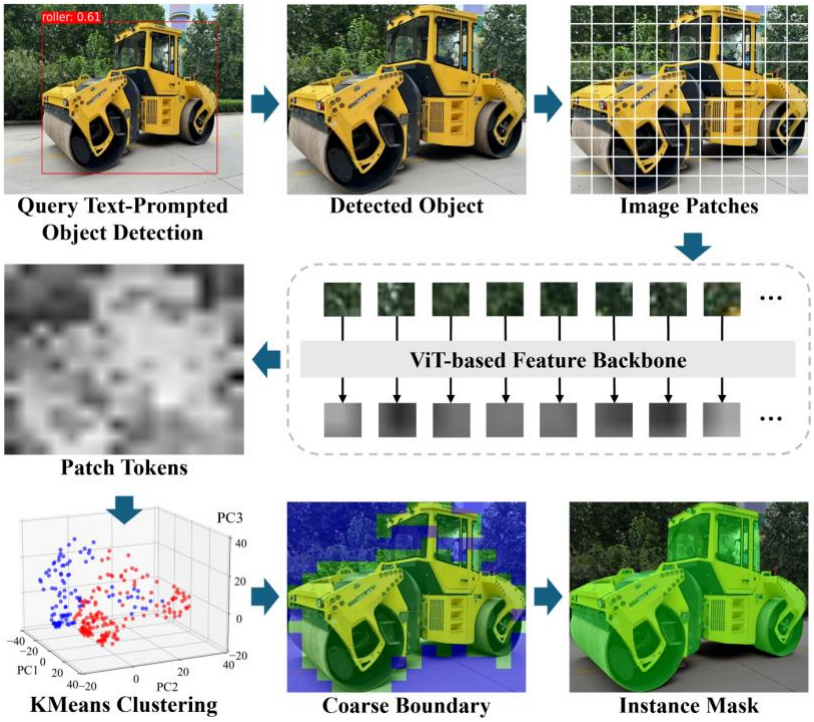


*Figure 2: Clustering-based mask generation process.*

As shown in Figure 2, we partition each detected object image into patches, which are then transformed into patch tokens. These patch tokens are clustered using KMeans into two regions, corresponding to the target object and background. Formally, given $n$ patch tokens ($x_1$, $x_2$, ..., $x_n$), KMeans aims to find two cluster centroids $\mu_1$ and $\mu_2$ that minimize the within-cluster sum of squared distances, $\min \sum \|x_i - \mu_k\|^2$, where each data point $x_i$ is assigned to cluster $C_k$ and $\mu_k$ denotes the centroid of $C_k$. The boundary between the clusters is used as a prompt for SAM, which refines the coarse boundary into a precise instance mask. Masks that are clearly inconsistent with the intended target category are manually removed, including 1) masks covering background regions or multiple objects, and 2) masks with incorrect or incomplete object representations. This lightweight filtering step serves as a quick quality control to ensure reliable pseudo labels for subsequent self-training, with minimal human effort involved. The correct masks are then used to bootstrap the subsequent self-training as described in the next section. In Figure 2, for visualization purposes, the high-dimensional patch tokens were projected to 3D using Principal Component Analysis (PCA).

## 3.3 Semi-supervised training data generation

A semi-supervised training model is proposed to iteratively transform the raw data into training data, as shown in Figure 3. During self-training, each detected object is cropped and treated as an individual training sample rather than using the entire image, preventing residual medium-confidence objects from remaining in the training sample and potentially introducing noise. Detected objects are incorporated into the training set, and a new round of self-training commences. The self-training process iterates until no new objects are detected, progressively annotating most raw images.

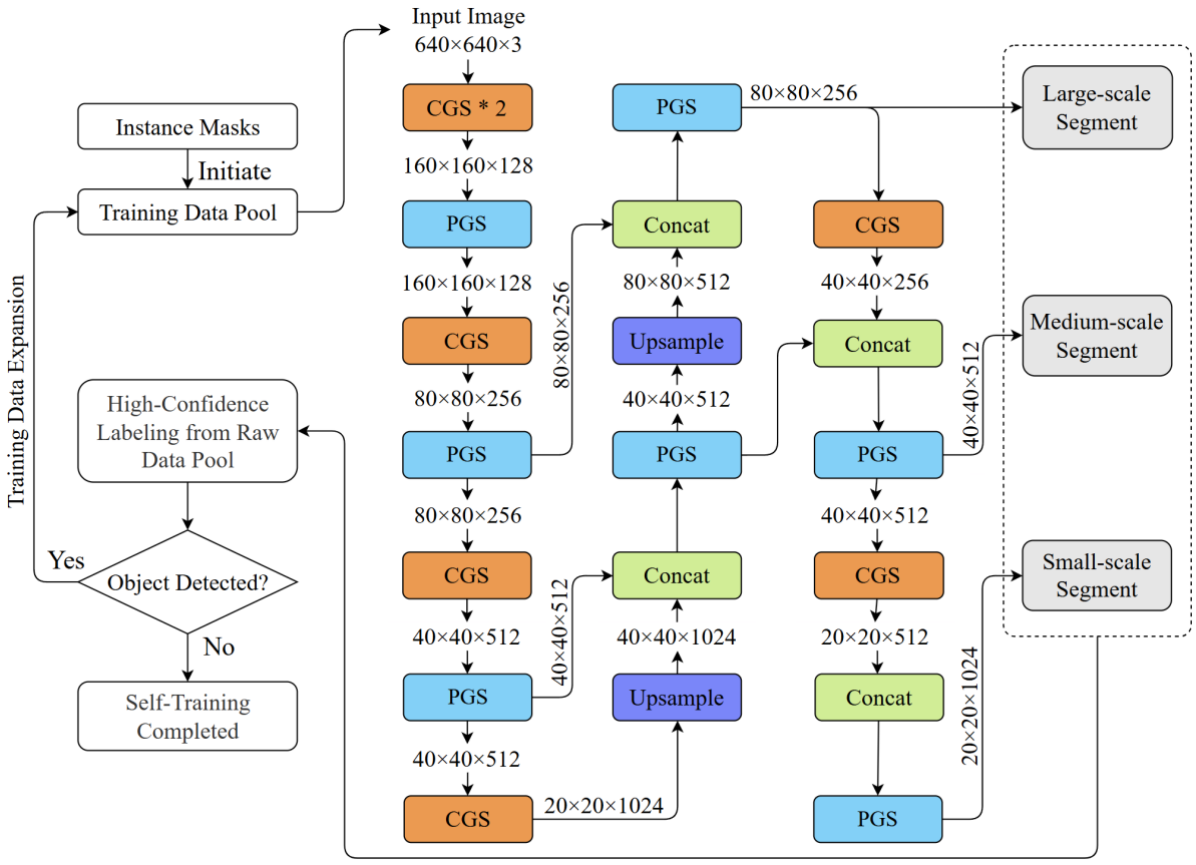


*Figure 3: Semi-supervised self-training model. The numbers between blocks (e.g., 160×160×128) indicate the size of the feature map (160×160) and the number of channels (128).*

As shown in Figure 3, the proposed model has two core blocks, CGS and PGS, which are specifically designed to improve training stability and feature robustness during iterative self-training. The CGS block, named after its three components, convolution, group normalization (GN), and SiLU activation, consists of a convolutional layer with a stride of 2 and padding of 1, followed by a GN layer and a SiLU activation function. The proposed semi-supervised self-training begins with a small set of pseudo instance mask labels, which then gradually generalizes to unlabeled data. During self-training, the generalization performance at each iteration is crucial, as it directly affects the effectiveness of label propagation. Studies have shown that small-batch training improves generalization while reducing memory consumption (Masters & Luschi, 2018). Consequently, GN was used instead of batch normalization (BN), as replacing BN with GN may stabilize the model across a wide range of batch sizes (Wu & He, 2020), making it particularly suitable for small-batch self-training.

The PGS block, named after its core components, partial convolution (PConv), GN, and the SiLU, is illustrated in Figure 4. In this block, input features are first processed by a CGS block for stable extraction, then split into two branches. One branch serves as a shortcut preserving original features, while the other passes through a partial convolution layer that convolves a subset of channels and transmits the rest directly. This design is particularly important in the early stages of self-training, where pseudo labels may still contain noise. By preserving part of the original feature information, the PGS block helps suppress error amplification and improves robustness to noisy pseudo labels. The partial convolution output is further processed by a pointwise convolution, followed by GN and SiLU activation to enhance nonlinearity. The main and shortcut branches are fused via a residual connection, integrating both original and transformed features. After multiple CGS and PGS processing steps, the final segment blocks operate at three scales (small, medium, and large) mirroring the multi-scale design concept of YOLO.

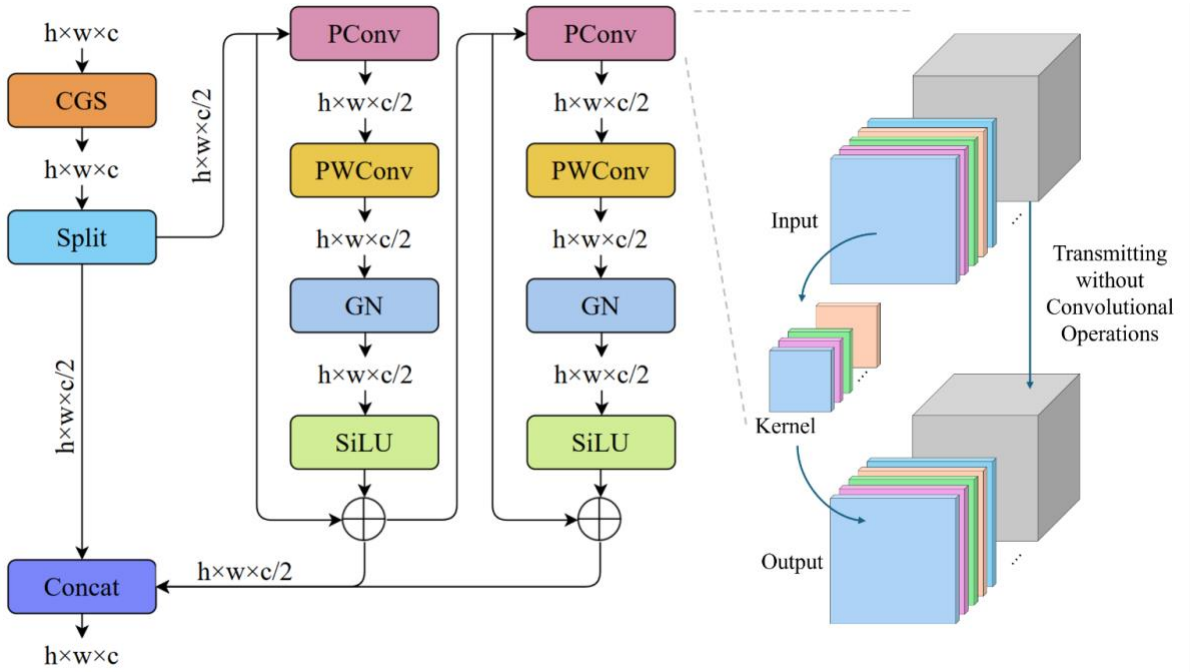


*Figure 4: Structure of the PGS block. PWConv denotes pointwise convolution; h and w denote height and width of feature map; c denotes number of channels; '+' denotes residual connection.*

# 4. EXPERIMENTAL SETUP

## 4.1 Datasets and implementation details

Experiments were conducted on the Construction Instance Segmentation (CIS) dataset (Yan et al., 2023), containing 50,000 images and 104,021 annotated instances across ten construction-specific categories. CIS was chosen because it is open-source, large-scale, and highly diverse, with images collected from 74 construction sites using multiple devices, providing a robust basis for evaluating generalization.

For model development and training, all implementations were based on the PyTorch framework. The experiments were performed on a high-performance server equipped with an NVIDIA A100 GPU with 80 GB of memory. The training was conducted with a batch size of 8 for 50 epochs, including 2 warm-up epochs, using the SGD optimizer with a learning rate of 0.00025. These hyperparameters were determined through preliminary tuning experiments to ensure stable convergence. The number of epochs was set to 50 based on convergence analysis, as performance gains became marginal beyond this point, while longer training increased the risk of overfitting. In addition, 2 warm-up epochs were introduced to stabilize optimization in the early stage of training.

Specifically, the confidence threshold was varied from 0.75 to 0.95 with a step size of 0.05 to evaluate its impact on the quantity and quality of generated training samples. Lower thresholds increased the number of samples but introduced more noisy labels, whereas higher thresholds improved label quality at the expense of data utilization. A threshold of 0.9 was selected as a balance during self-training and was later reduced to 0.5 during inference to improve recall.

## 4.2 Experimental procedures and evaluation metrics

Three sets of experiments were conducted. Corresponding procedures are summarized as follows. Experiment 1: Training data generation efficiency was evaluated using the CIS training set as the raw pool. Data were generated with TDG-CIS, manual annotation, and Grounded SAM for comparison. The objective of this experiment was to evaluate the training data generation efficiency of different approaches. Experiment 2: The effectiveness of generated training data was evaluated using two experimental settings. First, TDG-CIS-generated data from the CIS training set were compared with manually labeled CIS data derived from the same raw data pool to evaluate

the effectiveness of the proposed method under a comparable data scale. Second, TDG-CIS-generated data from a substantially larger web-collected pool were used to further assess the impact of data scale on model performance and demonstrate the scalability advantage of the proposed framework. Experiment 3: Ablation study evaluated the contribution of CGS and PGS blocks by removing each individually or both from TDG-CIS. The objective of this experiment was to evaluate the contribution of CGS and PGS to training data generation efficiency and the instance segmentation capability of TDG-CIS. Experiment 4: A sensitivity analysis was conducted to evaluate the impact of key hyperparameters on model performance.

The evaluation metrics used to assess the performance of the proposed method are as follows. DUR (Data Utilization Rate) measures the efficiency of converting raw images into training data and is calculated as the number of correct predictions divided by the number of ground-truth (GT) instances. mDUR (Mean DUR) represents the average DUR across all object categories and is obtained by dividing DUR by the number of object categories. AP (Average Precision) evaluates precision for a single class, indicating the proportion of correct predictions among predicted instances, and is calculated as the area under the precision-recall (PR) curve for that class. mAP (Mean AP) averages AP across all object categories. AR (Average Recall) measures recall for a single class, indicating the proportion of ground-truth instances correctly detected, calculated as the number of true positives divided by the number of GT instances. mAR (Mean AR) averages AR across all object categories.

## 5. RESULTS AND DISCUSSION

### 5.1 Experiment 1: Evaluation of training data generation efficiency

We first applied the proposed clustering-based mask generation method to process images drawn from a raw data pool constructed from the CIS training set, with examples illustrated in Figure 5, which presents both correctly generated and removed masks. On average, only 2.20% of masks were removed, with an inspection time of approximately 2 minutes per category (0.395 seconds per image), indicating that this step only requires a quick visual check. Overall, the process involves minimal human effort while maintaining a high level of automation and ensuring the quality of pseudo labels.

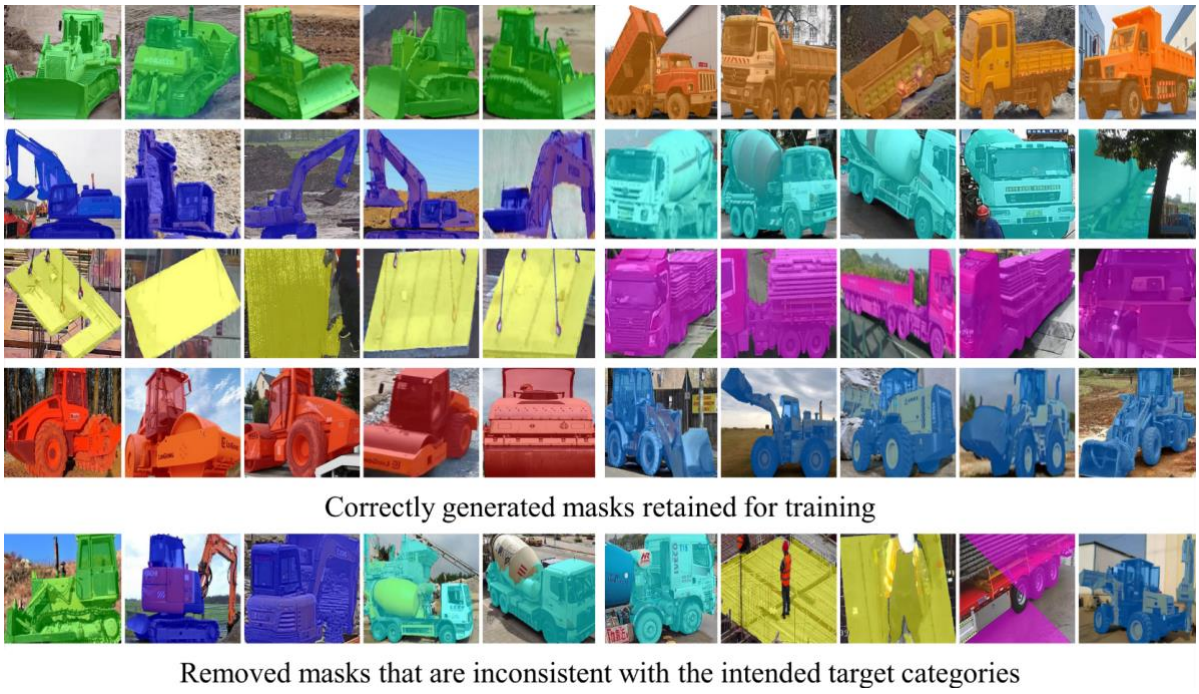


*Figure 5: Examples of the generated instance masks.*

The instance masks initialized self-training on the raw data pool. Figure 6 shows the training loss curves and the evolution of the model's confidence distribution across iterations, with legend numbers indicating iteration indices (00 for initial training). Figure 7 illustrates the evolution of the Wasserstein Distance (WD) (Lv et al., 2024) across iterations for each category, reflecting the distributional changes of the model's confidence scores (confidence > 0.5). The WD exhibits fluctuations rather than a monotonic decrease. Therefore, rather than using a single WD value, the self-training process is considered stable when the WD variation remains below a small threshold (e.g., 0.001) for three consecutive iterations, indicating negligible distributional changes. At this stage, only a few new training samples are added, so continuing self-training provides minimal benefit while increasing computational cost. This criterion is thus used to terminate self-training. During the self-training process, training data generation is accomplished simultaneously. Its efficiency, compared with the state-of-the-art Grounded SAM (Ren et al., 2024), which also does not require a source training domain, is reported in Table 1. TDG-CIS performs training data generation with high confidence during self-training. However, while a high confidence threshold improves prediction reliability, it inevitably leads to missed detections. To capture this trade-off, we adopt DUR, where a higher DUR indicates more generated training data.

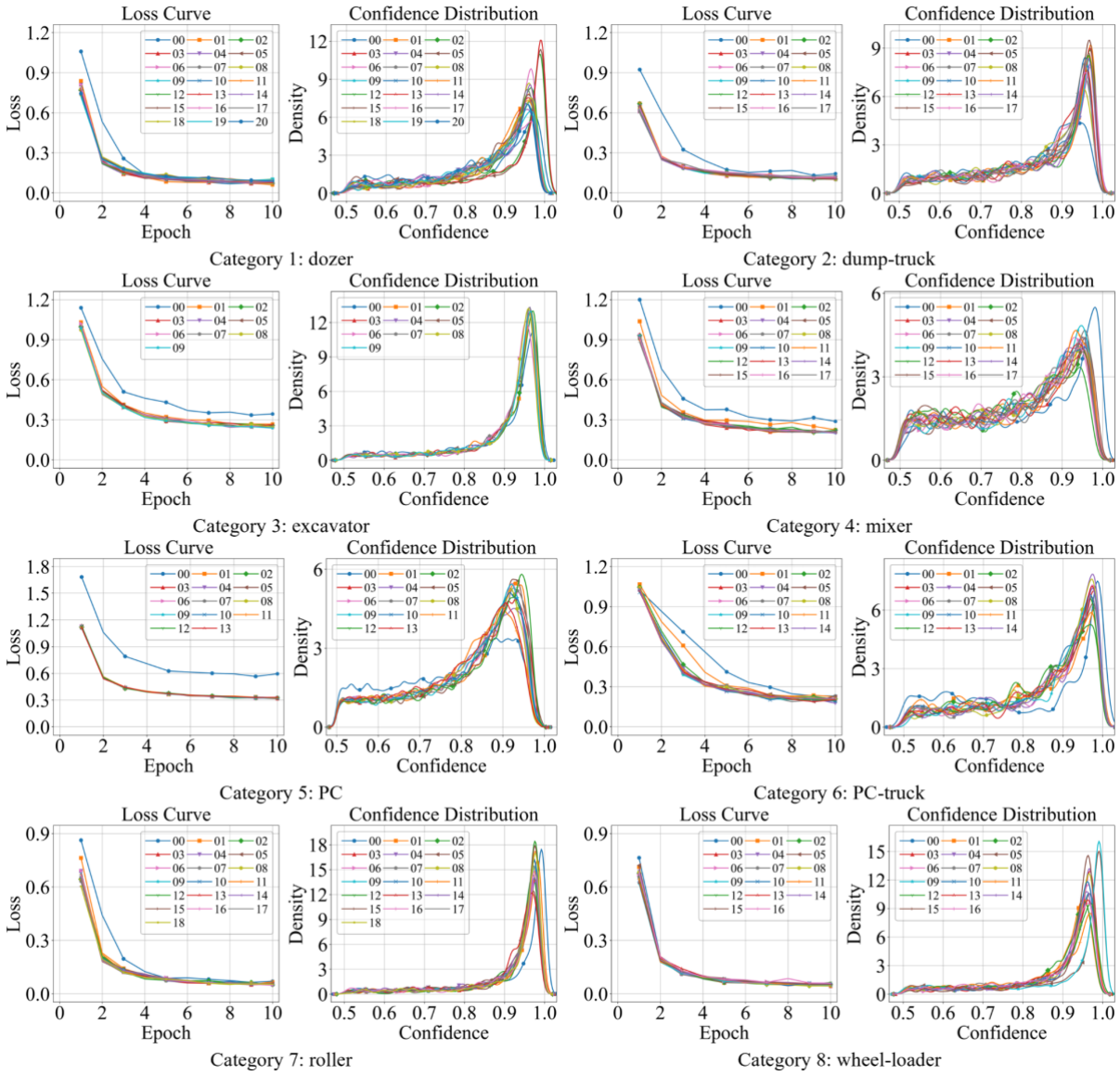


*Figure 6: Loss curves and confidence distributions in self-training.*

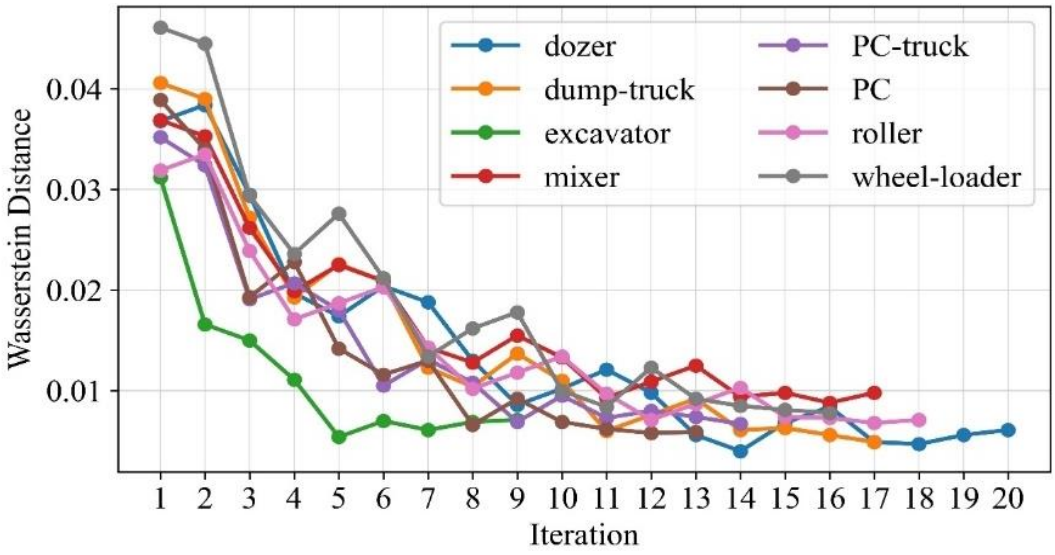


*Figure 7: Evolution of WD across iterations.*

*Table 1: Evaluation of training data generation efficiency. Notes: Grounded SAM uses default parameters (e.g., BOX_THRESHOLD = 0.25, TEXT_THRESHOLD = 0.25, NMS_THRESHOLD = 0.8).*

| Object Category | Num of GT | Num of Correct Predictions | | DUR (%) | |
|---|---|---|---|---|---|
| | | TDG-CIS | Grounded SAM | TDG-CIS | Grounded SAM |
| dozer | 1221 | 1065 | 402 | 87.2 | 32.9 |
| dump-truck | 1333 | 999 | 345 | 75.0 | 25.9 |
| excavator | 1367 | 1193 | 725 | 87.3 | 53.0 |
| mixer | 1161 | 857 | 517 | 73.8 | 44.5 |
| PC | 11105 | 6954 | 6108 | 62.6 | 55.0 |
| PC-truck | 620 | 507 | 230 | 81.8 | 37.1 |
| roller | 1091 | 832 | 307 | 76.3 | 28.1 |
| wheel-loader | 1310 | 1041 | 616 | 79.5 | 47.0 |
| Total/Mean | 19208 | 13448 | 9248 | 77.9 | 40.4 |

As noted in Section 2.2, existing non-fully-supervised and domain adaptation models for construction site analysis generally rely on source training domain, whereas TDG-CIS operates without it. Moreover, most such methods focus on object detection and semantic segmentation, while TDG-CIS addresses the more complex task of instance segmentation. Considering these differences, Grounded SAM is used as a practical reference baseline under a source-free setting. It should be noted that this comparison is not fully controlled, as Grounded SAM is applied as a general-purpose pre-trained model without explicit domain adaptation, whereas TDG-CIS performs iterative self-training on unlabeled construction-specific target-domain images. Therefore, the comparison mainly demonstrates the practical benefit of target-domain data generation over direct application of a general-purpose model. Results show that TDG-CIS achieves higher DUR than Grounded SAM across all categories, even though Grounded SAM has been pre-trained on a large-scale general-purpose dataset. This result highlights the importance of domain-specific data generation in enhancing segmentation performance and once again underscores the challenge posed by data distribution shifts across domains.

## 5.2 Experiment 2: Evaluation of effectiveness of generated training data

In this experiment, three training datasets (Table 2) were used to train the same baseline instance segmentation model (YOLOv11) to evaluate the effectiveness of automatically generated data and assess the impact of data scale. Specifically, Dataset 1 used TDG-CIS-generated data from the CIS training set, Dataset 2 used the manually labeled CIS training set, and Dataset 3 used TDG-CIS-generated data from over 70,000 de-duplicated images collected via a web-based monitoring platform (Figure 8). Figure 9 shows the loss curves of the baseline model.

*Table 2: Numbers of images and instances in the training datasets. Notes: Dataset 1: TDG-CIS-generated data from the images in CIS training set. Dataset 2: Manually labeled training set in CIS. Dataset 3: TDG-CIS-generated data from the web-based monitoring platform.*

| Object Category | Image Number | | | Instance Number | | |
|---|---|---|---|---|---|---|
| | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 1 | Dataset 2 | Dataset 3 |
| dozer | 1065 | 1208 | 2112 | 1065 | 1221 | 2125 |
| dump-truck | 999 | 1169 | 2228 | 999 | 1333 | 2410 |
| excavator | 1148 | 1188 | 2432 | 1193 | 1367 | 2686 |
| mixer | 857 | 1061 | 1990 | 857 | 1161 | 2088 |
| PC | 4242 | 2016 | 5738 | 6954 | 11105 | 15060 |
| PC-truck | 506 | 617 | 1173 | 507 | 620 | 1182 |
| roller | 832 | 1066 | 1857 | 832 | 1091 | 1857 |
| wheel-loader | 1041 | 1296 | 2150 | 1041 | 1310 | 2166 |
| Total | 10690 | 9621 | 19680 | 13448 | 19208 | 29304 |

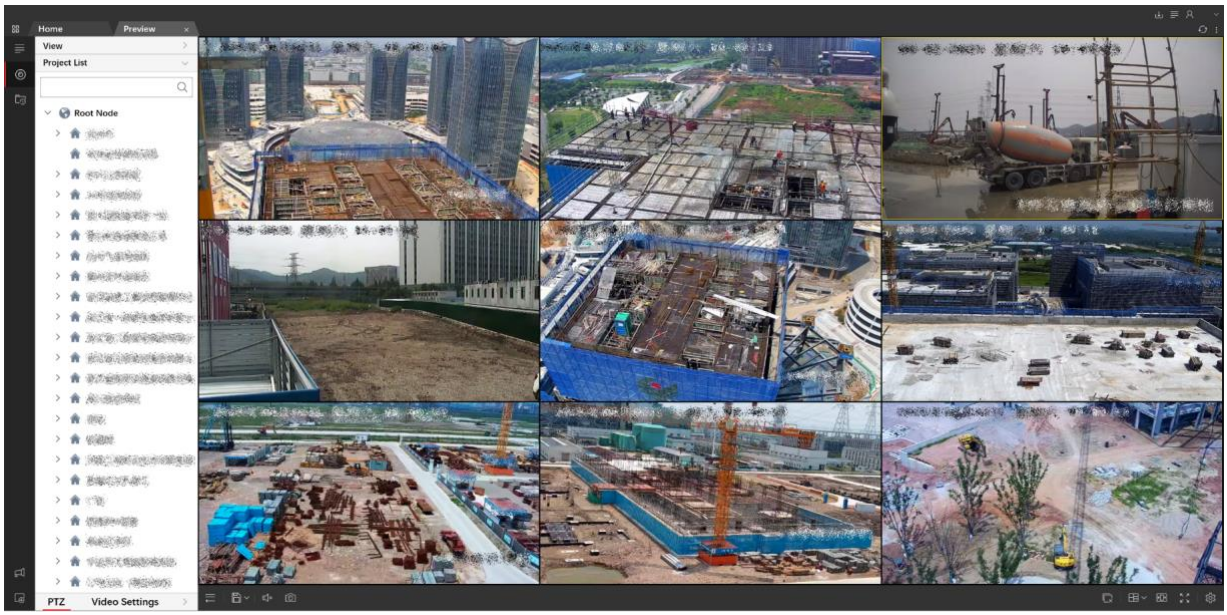


*Figure 8: Web-based on-site monitoring platform for larger-scale raw data collection.*

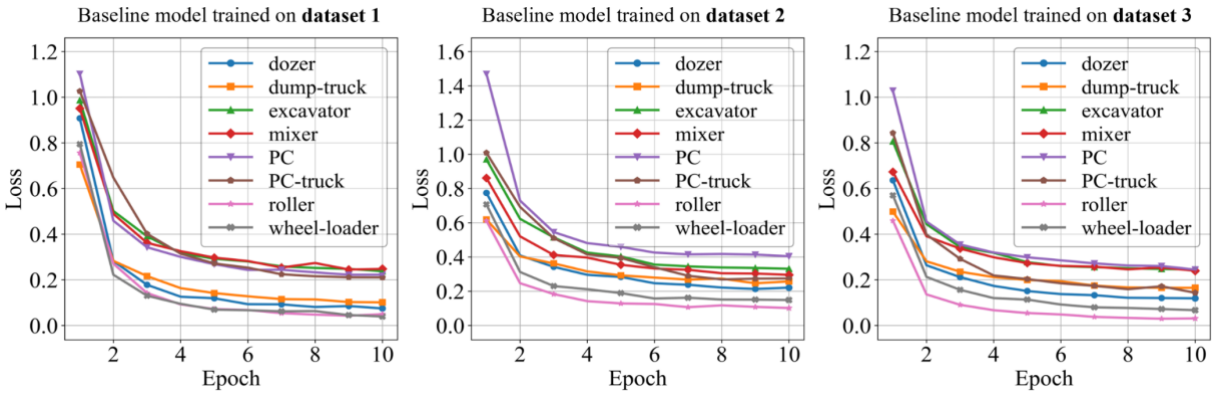


*Figure 9: Loss curves of the same baseline model trained on different training datasets.*

All the trained models were evaluated on the CIS test set to examine the impact of data scale. The results are summarized in Table 3, with PR curves in Figure 10.

*Table 3: Evaluation of the same baseline model trained on different datasets. Notes: Dataset 1: TDG-CIS-generated data from the images in CIS training set. Dataset 2: Manually labeled training set in CIS. Dataset 3: TDG-CIS-generated data from the web-based monitoring platform.*

| Object Category | AP (%) | | | AR (%) | | |
|---|---|---|---|---|---|---|
| | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 1 | Dataset 2 | Dataset 3 |
| dozer | 93.9 | 96.6 | 98.5 | 89.6 | 94.4 | 94.3 |
| dump-truck | 82.0 | 87.8 | 88.0 | 75.6 | 82.2 | 82.1 |
| excavator | 91.1 | 95.1 | 98.1 | 85.0 | 88.7 | 94.4 |
| mixer | 71.2 | 86.5 | 87.5 | 66.7 | 78.1 | 77.3 |
| PC | 87.9 | 91.1 | 93.8 | 79.6 | 83.0 | 86.6 |
| PC-truck | 91.8 | 96.7 | 98.9 | 84.9 | 92.5 | 97.3 |
| roller | 93.0 | 97.0 | 97.0 | 82.4 | 91.8 | 91.3 |
| wheel-loader | 84.2 | 93.0 | 92.9 | 73.1 | 82.9 | 85.5 |
| Mean | 86.8 | 92.9 | 94.3 | 79.6 | 86.7 | 88.6 |

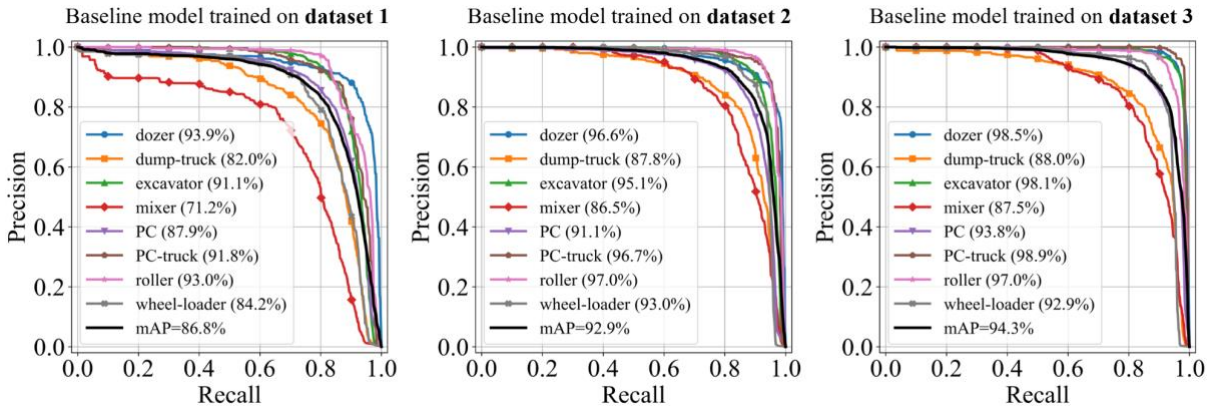


*Figure 10: PR curves of the same baseline model trained on different training datasets.*

Both Table 3 and Figure 10 show that within the same basic pool, the baseline model trained on manual annotations (Dataset 2) outperform those trained on TDG-CIS data (Dataset 1), likely due to a higher data utilization rate (mDUR 100% vs. 77.9%). However, with the larger-scale pool, TDG-CIS (Dataset 3) surpasses manual data (Dataset 2) in mAP and mAR (94.3% vs. 92.9% and 88.6% vs. 86.7%), demonstrating that scaling the raw pool enhances TDG-CIS's ability to generate diverse training samples. Example inferences in Figure 11 confirm the enhanced performance of the baseline model trained on TDG-CIS-generated data from the larger-scale pool.

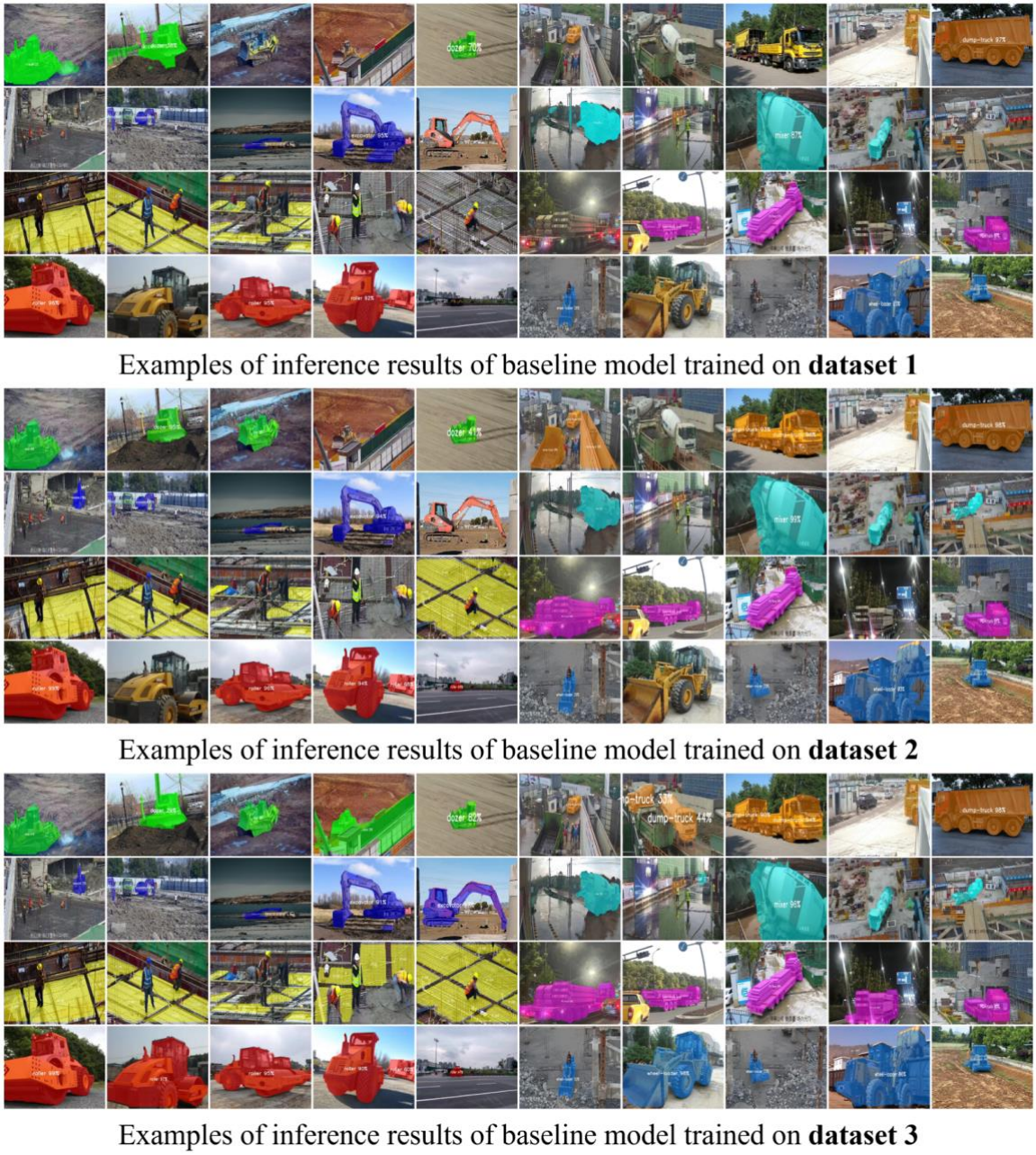


*Figure 11: Example inference results of the same baseline model trained on different datasets.*

## 5.3 Experiment 3: Ablation study on model blocks

To assess the contributions of CGS and PGS, TDG-CIS was tested under three ablation settings: removing each individually or both. Table 4 summarizes the results of the ablation study, and Figure 12 shows the corresponding PR curves.

*Table 4: Results of ablation study on CGS and PGS.*

| Object Category | TDG-CIS Full | | | TDG-CIS w/o CGS | | | TDG-CIS w/o PGS | | | TDG-CIS w/o CGS & PGS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DUR (%) | AP (%) | AR (%) | DUR (%) | AP (%) | AR (%) | DUR (%) | AP (%) | AR (%) | DUR (%) | AP (%) | AR (%) |
| dozer | 87.2 | 95.0 | 90.2 | 86.8 | 94.9 | 91.3 | 86.3 | 93.9 | 91.0 | 85.9 | 93.6 | 90.4 |
| dump-truck | 75.0 | 82.1 | 75.2 | 74.6 | 81.4 | 76.1 | 74.1 | 81.1 | 74.4 | 73.8 | 81.1 | 72.3 |
| excavator | 87.3 | 91.6 | 85.5 | 86.9 | 91.3 | 82.8 | 86.5 | 91.1 | 85.0 | 86.1 | 90.3 | 84.0 |
| mixer | 73.8 | 71.8 | 68.2 | 73.4 | 71.2 | 65.8 | 72.9 | 70.4 | 68.5 | 72.5 | 63.6 | 71.6 |
| PC | 62.6 | 86.3 | 78.6 | 62.2 | 85.7 | 78.6 | 61.8 | 85.3 | 77.2 | 61.5 | 85.6 | 77.6 |
| PC-truck | 81.8 | 91.8 | 84.9 | 81.4 | 91.7 | 85.8 | 81.0 | 91.6 | 87.7 | 80.6 | 91.5 | 81.6 |
| roller | 76.3 | 94.8 | 85.1 | 75.9 | 94.5 | 84.6 | 75.4 | 94.1 | 81.5 | 75.0 | 94.0 | 83.4 |
| wheel-loader | 79.5 | 87.3 | 81.1 | 79.1 | 85.9 | 76.1 | 78.7 | 83.5 | 73.5 | 78.3 | 85.2 | 75.2 |
| Mean | 77.9 | 87.5 | 81.1 | 77.5 | 87.0 | 80.1 | 77.1 | 86.3 | 79.9 | 76.7 | 85.6 | 79.5 |

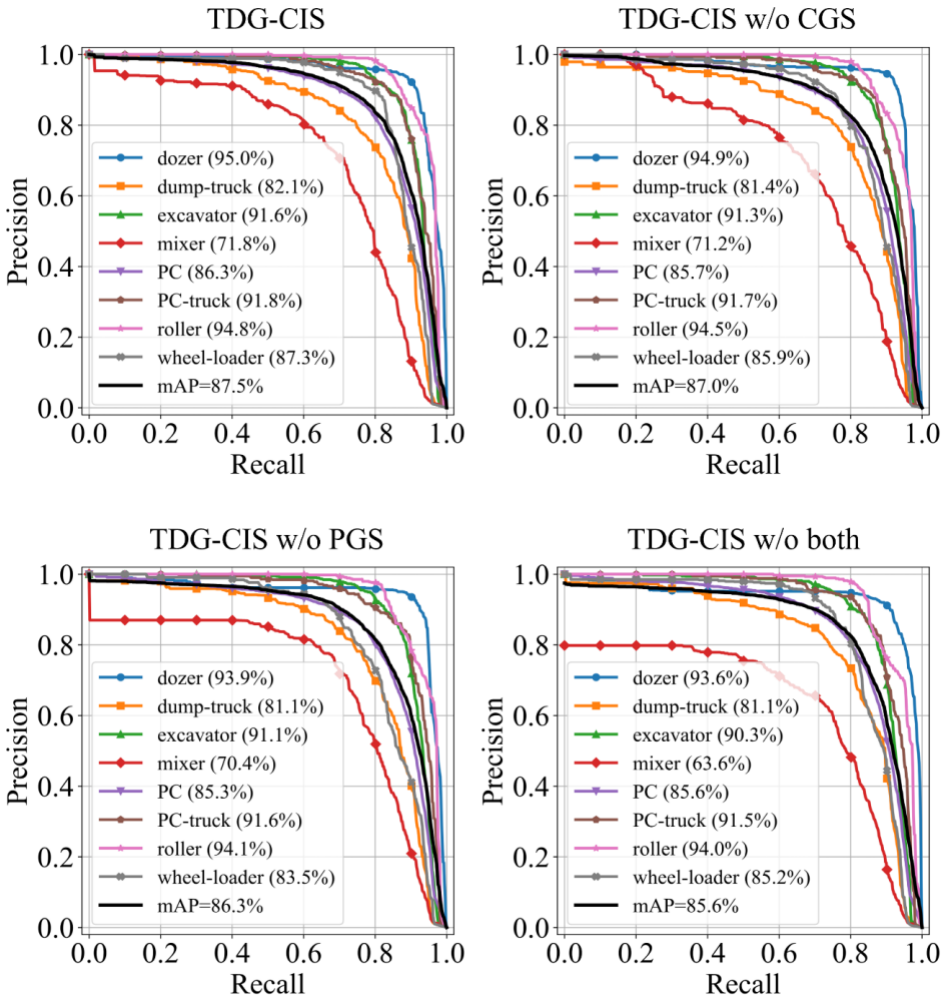


*Figure 12: PR curves of the TDG-CIS and its ablated versions.*

DUR metrics show that removing CGS or PGS slightly lowers mDUR from 77.9% to 77.5% and 77.1%, respectively, while removing both drops it to 76.7%, indicating both blocks enhance data utilization.

In terms of detection performance, measured by AP and AR, similar trends are observed. The full TDG-CIS model achieves the highest mAP (87.5%) and mAR (81.1%). Ablating the CGS or PGS individually results in minor drops in mAP (87.0% and 86.3%) and mAR (80.1% and 79.9%), while removing both blocks simultaneously leads to more pronounced decreases (mAP = 85.6%, mAR = 79.5%). This pattern highlights that both CGS and PGS contribute positively to the overall detection accuracy and recall. Ablation results confirm that the proposed architectural enhancements improve data utilization and instance segmentation performance across all categories in CIS.

## 5.4 Sensitivity analysis of key hyperparameters

To further evaluate the robustness of the proposed framework, a sensitivity analysis was conducted on the confidence threshold, which is a key hyperparameter in the self-training process. The threshold was varied from 0.75 to 0.95 with a step size of 0.05, and its impact on data utilization and segmentation performance was analyzed. The detailed results are summarized in Figure 13.

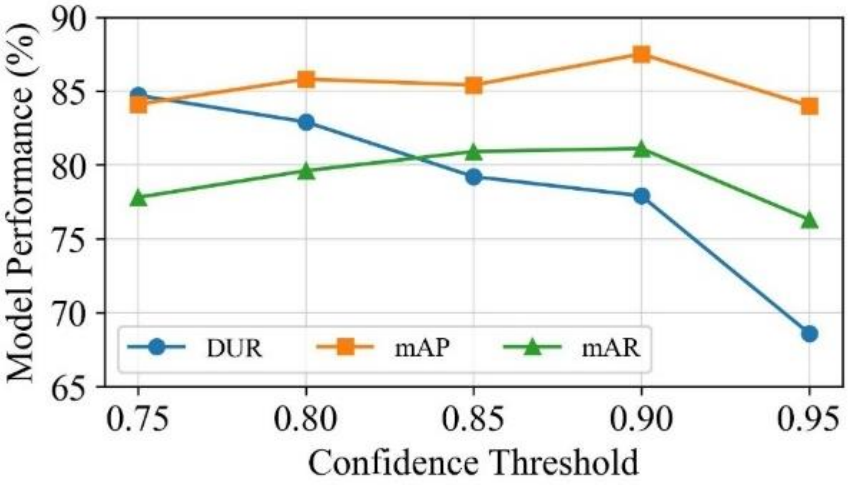


*Figure 13: Model sensitivity to confidence threshold.*

As the confidence threshold decreases from 0.95 to 0.75, DUR increases from 72.6% to 84.7%, as more generated samples are retained. However, this also introduces more noisy labels, leading to a decline in segmentation performance, with mAP decreasing from 86.0% to 84.1% and mAR from 80.3% to 77.8%. Conversely, increasing the threshold improves label reliability but reduces the number of usable samples. Notably, when the threshold is set to 0.95, the initial number of generated samples is significantly reduced, which limits the diversity and scale of the training data and results in decreased segmentation performance compared to the optimal setting.

Overall, the results demonstrate a clear trade-off between data quantity and label quality. Among the tested values, a threshold of 0.9 achieves the best balance, yielding the highest mAP (87.5%) and mAR (81.1%) while maintaining relatively high data utilization (77.9%). Furthermore, the performance remains relatively stable across a reasonable range of threshold values, indicating that the proposed framework is not overly sensitive to this hyperparameter.

## 5.5 Research implications and limitations

The experimental results highlight the notable advantages of the proposed TDG-CIS approach and its significant implications for computer vision applications in construction. With a data utilization rate of 77.9%, TDG-CIS is able to convert a substantial proportion of unlabeled target-domain site images into usable instance-level training data without relying on source-domain supervision, demonstrating its effectiveness in alleviating data scarcity and cross-domain distribution shift challenges. The results in Experiment 2 further provide stronger evidence for this implication: under the same raw data pool, the baseline model trained on TDG-CIS-generated data achieved

competitive performance compared with the manually annotated counterpart, while the performance further improved when the raw data pool was substantially expanded, reaching 94.3% mAP and 88.6% mAR. This trend suggests that the proposed framework particularly benefits from large-scale unlabeled data, where its automatic data generation capability can be fully leveraged.

The findings also indicate that the main advantage of TDG-CIS lies in its scalability. Unlike manual annotation, whose cost increases almost linearly with data volume, the proposed framework allows model performance to continue improving as additional unlabeled target-domain data become available. This characteristic is especially valuable in construction monitoring, where continuous image streams are routinely collected from cameras, drones, and on-site platforms. Overall, this study provides a scalable pathway for target-domain training data generation and offers practical implications for deploying high-performance vision models in intelligent construction.

Nevertheless, several limitations should be acknowledged. First, although the proposed framework is highly automated, it still requires lightweight human verification during initialization to remove inconsistent masks. Second, a relatively high confidence threshold was employed during the self-training process to mitigate noise in the training data pool. Although this strategy helps maintain label quality, it cannot entirely eliminate the inclusion of noisy samples. Third, the current validation is limited to only eight construction-specific object categories. Fourth, the comparison with Grounded SAM should be interpreted as a practical reference under a source-free setting rather than a fully controlled comparison. Although both methods do not rely on source-domain supervision, Grounded SAM is applied as a general-purpose model without explicit domain adaptation, whereas TDG-CIS uses iterative self-training on unlabeled target-domain images. This difference limits the extent to which the comparison can evidence methodological superiority. Correspondingly, future research will focus on four directions. First, efforts will be made to further improve the degree of automation by reducing the need for human verification during initialization. Second, more robust noise-aware self-training strategies and adaptive confidence threshold mechanisms will be explored to further suppress noisy samples while maintaining data utilization. Third, the proposed framework will be extended and validated on a broader range of object categories and engineering scenarios to further assess its generalizability and scalability. Fourth, more directly comparable source-free, construction-specific instance segmentation baselines will be included when such methods become available or can be adapted under the same experimental setting.

## 6. CONCLUSIONS

This study addresses the challenge of performance degradation in construction instance segmentation due to training data scarcity and domain distribution shifts in target domains. The research aims to enable high-quality training data generation from unlabeled target-domain images without relying on source training domains. To achieve this, we propose TDG-CIS, a clustering-initialized semi-supervised method that integrates KMeans clustering on image patch tokens for initial mask generation and a novel self-training framework for iterative refinement.

Experimental results demonstrate that TDG-CIS achieves a 77.9% data utilization rate, 87.5% mAP, and 81.1% mAR on a large-scale public dataset covering over 70 construction domains. Notably, models trained on TDG-CIS-generated data outperform those trained on manually labeled data, with mAP increasing from 92.9% to 94.3% and mAR from 86.7% to 88.6%.

The key contributions of this work are twofold: (1) we propose a clustering-initialized semi-supervised training data generation method for construction instance segmentation (TDG-CIS), which enables high-quality instance-level labeling and supports model training and inference in unlabeled target domains without relying on source training domains; and (2) TDG-CIS achieves superior data utilization and model performance on a large-scale benchmark encompassing over 70 construction domains. This work advances scalable computer vision solutions for intelligent construction monitoring under real-world data distribution shifts across target domains.

## ACKNOWLEDGEMENTS

This research was supported by the National Natural Science Foundation of China (No. 72201247 and No. 72304098). The authors would like to acknowledge Zhejiang Construction Investment Group Co., Ltd. for visual data access.

## REFERENCES


Amini, M., Feofanov, V., Pauletto, L., Hadjadj, L., Devijver, E., & Maximov, Y. (2025). Self-training: A survey. *Neurocomputing*, 616, 128904. https://doi.org/10.2139/ssrn.4875054

An, X., Li, Z., Zuguang, L., Chengzhi, W., Pengfei, L., & Zhiwei, L. (2021). Dataset and benchmark for detecting moving objects in construction sites. *Automation in Construction*, 122, 103482. https://doi.org/10.1016/j.autcon.2020.103482

Barrera-Animas, Yair, A., Delgado, D., & Manuel, J. (2023). Generating real-world-like labelled synthetic datasets for construction site applications. *Automation in Construction*, 151, 104850. https://doi.org/10.1016/j.autcon.2023.104850

Bock, T. (2015). The future of construction automation: Technological disruption and the upcoming ubiquity of robotics. *Automation in Construction*, 59, 113-121. https://doi.org/10.1016/j.autcon.2015.07.022

Chern, W.-C., Kim, T., Nguyen, T. V., Asari, V. K., & Kim, H. (2023). Self-supervised sub-category exploration for pseudo label generation. *Automation in Construction*, 151, 104862. https://doi.org/10.1016/j.autcon.2023.104862

Deng, X., Yu, Q., Wang, P., Shen, X., & Chen, L. C. (2024). COCONut: Modernizing COCO Segmentation. *Proceedings of the 2024 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr52733.2024.02065

Ghelmani, A., & Hammad, A. (2023). Self-supervised contrastive video representation learning for construction equipment activity recognition on limited dataset. *Automation in Construction*, 154, 105001. https://doi.org/10.1016/j.autcon.2023.105001

Gupta, A., Dollar, P., & Girshick, R. (2019). LVIS: A Dataset for Large Vocabulary Instance Segmentation. *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2019.00550

Gupta, M., Wei, C., & Czerniawski, T. (2024). Semi-supervised symbol detection for piping and instrumentation drawings. *Automation in Construction*, 159, 105260. https://doi.org/10.1016/j.autcon.2023.105260

Hong, Y., Chern, W.-C., Nguyen, T. V., Cai, H., & Kim, H. (2023). Semi-supervised domain adaptation for segmentation models on different monitoring settings. *Automation in Construction*, 149. https://doi.org/10.1016/j.autcon.2023.104773

Huang, Q., Wang, J., Song, Y., Cui, W., Li, H., Wang, S., Dai, P., Zhao, X., & Li, Q. (2024). Synthetic-to-realistic domain adaptation for cold-start of rail inspection systems. *Computer-Aided Civil and Infrastructure Engineering*, 39(3), 424-437. https://doi.org/10.1111/mice.13087

Huang, Y., Yang, H., Sun, K., Zhang, S., Cao, L., Jiang, G., & Ji, R. (2023). Interformer: Real-Time Interactive Image Segmentation. *Proceedings of the 2023 IEEE International Conference on Computer Vision*. https://doi.org/10.1109/iccv51070.2023.02038

Kim, H.-S., Seong, J., & Jung, H.-J. (2023a). Real-time struck-by hazards detection system for small- and medium-sized construction sites based on computer vision using far-field surveillance videos. *Journal of Computing in Civil Engineering*, 37(6), 04023028. https://doi.org/10.1061/jccee5.cpeng-5238

Kim, H., Kim, H., Hong, Y. W., & Byun, H. (2018). Detecting construction equipment using a region-based fully convolutional network and transfer learning. *Journal of Computing in Civil Engineering*, 32, 04017082. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000731

Kim, H. S., Seong, J., & Jung, H. J. (2024a). Optimal domain adaptive object detection with self-training and adversarial-based approach for construction site monitoring. *Automation in Construction*, 158, 105244. https://doi.org/10.1016/j.autcon.2023.105244

Kim, J., Kim, D., Lee, S., & Chi, S. (2023b). Hybrid DNN training using both synthetic and real construction images to overcome training data shortage. *Automation in Construction*, 149, 104771. https://doi.org/10.1016/j.autcon.2023.104771

Kim, J., Wang, I., & Yu, J. (2024b). Experimental study on using synthetic images as a portion of training dataset for object recognition in construction site. *Buildings*, 14(5), 1-14. https://doi.org/10.3390/buildings14051454

Kim, T., Chern, W.-C., Kim, S., Asari, V. K., & Kim, H. (2025). Moving-feature-driven label propagation for training data generation from target domains. *Computers in Industry*, 171, 104335. https://doi.org/10.1016/j.compind.2025.104335

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W. Y., & Dollár, P. (2023). Segment Anything. *Proceedings of the 2023 IEEE International Conference on Computer Vision*. https://doi.org/10.48550/arXiv.2304.02643

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436-444. https://doi.org/10.1038/nature14539

Liang, Y., Chen, G., Li, S., & Xu, Z. (2023). Intelligent defect diagnosis of appearance quality for prefabricated concrete components based on target detection and multimodal fusion decision. *Journal of Computing in Civil Engineering*, 37(6), 04023032. https://doi.org/10.1061/jccee5.cpeng-5460

Liu, Q., Xu, Z., Bertasius, G., & Niethammer, M. (2023). SimpleClick: Interactive Image Segmentation with Simple Vision Transformers. *Proceedings of the 2023 IEEE International Conference on Computer Vision*. https://doi.org/10.1109/iccv51070.2023.02037

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., & Zhang, L. (2024). Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *Proceedings of the 2024 European Conference on Computer Vision*. https://doi.org/10.1007/978-3-031-72970-6_3

Lv, J., Yang, H., & Li, P. (2024). Wasserstein Distance Rivals Kullback-Leibler Divergence for Knowledge Distillation. *Proceedings of the 2024 Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=1qfdCAXn6K

Masters, D., & Luschi, C. (2018). Revisiting small batch training for deepneural networks. *arXiv preprint*, 1-18. https://doi.org/10.48550/arXiv.1804.07612

Mei, X., Ma, W., Xu, F., & Zhang, Z. (2024). Vision-based detection of unsafe worker guardrail climbing based on posture and instance segmentation data fusion. *Journal of Construction Engineering and Management*, 150(11), 04024156. https://doi.org/10.1061/jcemd4.Coeng-14266

Midwinter, M., Al-Sabbag, Z. A., & Yeum, C. M. (2023). Unsupervised defect segmentation with pose priors. *Computer-Aided Civil and Infrastructure Engineering*, 38(17), 2455-2471. https://doi.org/10.1111/mice.13001

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2022). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7), 3523-3542. https://doi.org/10.1109/TPAMI.2021.3059968

Mondal, S. A., Bhattacharaya, T., Rai, A., Sodhi, G. S., Bansal, R., Mondal, A., & Gupta, A. (2025). AutoTag: A framework for generating training data. *Progress in Artificial Intelligence*, 14(2), 191-210. https://doi.org/10.1007/s13748-024-00360-x

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., & Bojanowski, P. (2024). Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 1, 1-32. https://doi.org/10.48550/arXiv.2304.07193

Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., & Zhang, L. (2024). Grounded SAM: Assembling open-world models for diverse visual tasks. *arXiv preprint*, 1-11. https://doi.org/10.48550/arXiv.2401.14159

Sun, T., Fan, Q., & Shao, Y. (2025). Deep learning-based rebar detection and instance segmentation in images. *Advanced Engineering Informatics*, 65, 103224. https://doi.org/10.1016/j.aei.2025.103224

Wang, G., Zhou, Y., & Cao, D. (2025). Artificial intelligence in construction: Topic-based technology mapping based on patent data. *Automation in Construction*, 172, 106073. https://doi.org/10.1016/j.autcon.2025.106073

Weng, X., Huang, Y., Li, Y., Yang, H., & Yu, S. (2023). Unsupervised domain adaptation for crack detection. *Automation in Construction*, 153, 104939. https://doi.org/10.1016/j.autcon.2023.104939

Wu, Y., & He, K. (2020). Group normalization. *International Journal of Computer Vision*, 128(3), 742-756. https://doi.org/10.1007/s11263-019-01198-w

Xu, S., Wang, J., Shou, W., Ngo, T., Sadick, A.-M., & Wang, X. (2020). Computer vision techniques in construction: A critical review. *Archives of Computational Methods in Engineering*, 28(5), 3383-3397. https://doi.org/10.1007/s11831-020-09504-3

Yan, X., Jin, R., Zhang, H., Gao, H., & Xu, S. (2025). Computer vision-based intelligent monitoring of disruptions due to construction machinery arrival delay. *Journal of Computing in Civil Engineering*, 39(3), 04025011. https://doi.org/10.1061/JCCEE5.CPENG-6178

Yan, X., Zhang, H., Wu, Y., Lin, C., & Liu, S. (2023). Construction Instance Segmentation (CIS) dataset for deep learning-based computer vision. *Automation in Construction*, 156, 105083. https://doi.org/10.1016/j.autcon.2023.105083

Yong, G., Jeon, K., Gil, D., & Lee, G. (2023). Prompt engineering for zero-shot and few-shot defect detection and classification using a visual-language pretrained model. *Computer-Aided Civil and Infrastructure Engineering*, 38(11), 1536-1554. https://doi.org/10.1111/mice.12954

Zha, D., Bhat, Z. P., Lai, K. H., Yang, F., Jiang, Z., Zhong, S., & Hu, X. (2025). Data-centric artificial intelligence: A survey. *ACM Computing Surveys*, 57(5), 1-42. https://doi.org/10.1145/3711118

Zhang, H., Qian, Z., Zhou, W., Min, Y., & Liu, P. (2024). A controllable generative model for generating pavement crack images in complex scenes. *Computer-Aided Civil and Infrastructure Engineering*, 39(12), 1795-1810. https://doi.org/10.1111/mice.13171