Less is More: Reinforcement Learning-powered Training Data Quality Evaluation and Sampling for Enhancing Computer Vision in Construction

Ziqing Wang[1] and Jinwoo Kim[2]

[1]School of Civil and Environmental Engineering, Nanyang Technological University, Singapore
[2]Department of Civil and Environmental Engineering, Hanyang University, Republic of Korea
ziqing003@e.ntu.edu.sg, jinwookim@hanyang.ac.kr

Abstract

Deep learning-based computer vision in construction has often assumed that more training data leads to better performance. This quantity-driven approach overlooks the detrimental effects of redundant, noisy, or mislabeled samples that can degrade model accuracy while inflating computational costs. To overcome this limitation, we propose and test an alternative hypothesis: selectively removing low-quality training samples can outperform indiscriminate use of all data. Specifically, we present a reinforcement learning (RL)-powered data quality evaluation framework that scores individual training images and prioritizes the most beneficial samples for model learning. Within a unified environment, the RL agent (referred to as the RL-evaluator) interacts with an object detector by probabilistically selecting subsets of high-quality samples for model training and using validation performance as a reward signal to optimize its selection policy. Experiments on a construction benchmark dataset of varying sizes show that removing low-quality samples identified by the RL-evaluator consistently improves detection performance compared to training on the full dataset, while also reducing computational overhead. These findings demonstrate the proposed method as a scalable and robust way for enhancing computer vision in construction.

Keywords –

Construction; Computer Vision; Reinforcement Learning; Data Quantity; Data Quality

1 Introduction

Deep learning-based computer vision has emerged as a key driver in the construction industry's shift toward digital project delivery and robotic automation. Instead of relying on manual site monitoring, which is often inconsistent and labor-intensive, automated computer vision systems can systematically interpret large volumes of visual data with greater speed and reliability. Such capabilities make it possible to detect and track resources [1], analyze ongoing activities [2], and identify operational irregularities [3], thereby supporting practical needs such as safety management [4], progress monitoring [5], and efficiency optimization [6]. Computer vision also provides the perceptual input required for robotics, equipping autonomous machines with spatial awareness and adaptability in unstructured environments. This connection between visual perception and autonomous operation further contributes to the development of digital twins, which integrate physical processes with virtual models. Taken collectively, these advances signify not only incremental improvements in construction monitoring and control but also a redefinition of construction practice toward intelligent, automated, and fully digitalized production systems.

Although computer vision in construction has advanced rapidly, much of the progress is based on the widely accepted hypothesis that larger datasets lead to better model performance [7]. This reliance on sheer data quantity has led to massive image collections covering a wide range of objects, activities, and site conditions, thereby enabling state-of-the-art benchmarks. However, model performance depends not only on dataset size but also on the quality of individual training samples. Redundant, noisy, and mislabeled data can introduce bias, slow or even prevent convergence, and even reduce accuracy. Meanwhile, data redundancy imposes heavy computational and storage demands. We therefore advance an alternative hypothesis: excluding some low-quality or harmful data may lead to superior results compared with using the full dataset. Despite its intuitive appeal, this hypothesis has rarely been systematically tested for computer vision in construction. Existing methods—such as leave-one-out and Shapley-based evaluation—are computationally prohibitive for high-

dimensional visual tasks, while active learning techniques typically rely on uncertainty measures, which do not directly reflect how each sample influences the learning dynamics or final model accuracy. As a result, the lack of scalable and reliable data quality evaluation strategies continues to constrain further improvements in the field.

To investigate our hypothesis, we present an RL-powered framework that evaluates the quality of individual training images and prioritizes those most valuable for model learning, while discarding samples deemed harmful or redundant. In this setup, the RL agent—referred to as the RL-evaluator and a computer vision model are embedded in an RL training environment. The RL-evaluator assigns quality scores (potential contributions for shaping model performance) to individual images and stochastically selects subsets of high-quality samples, which are then used to update the computer vision model. The model performance on a validation dataset is then used to provide the reward signal for refining the evaluator's policy. Through these repeated interactions, the RL-evaluator progressively learns a policy that guides sample selection toward improved model accuracy and data efficiency. We validate this framework on the Moving Objects in Construction Sites (MOCS) [7] dataset of different scales. We design and compare model performance using the following data curation strategies: (i) full data, (ii) low-quality removal, (iii) high-quality removal, and (iv) random removal. As an initial attempt, our experiments focus on worker detection, a fundamental and well-represented task in construction, allowing detailed investigation of the proposed framework.

This work contributes to advancing computer vision in construction in three aspects. First, we present an RL-powered framework for object detection tasks, where the training of a detection model is jointly integrated with the RL evaluator to enable fine-grained assessment of individual training samples. Second, we challenge the prevailing belief that larger datasets yield better model performance and instead demonstrate that removing low-quality samples identified by the RL evaluator can improve detection accuracy while reducing computational cost. Third, we show that the framework achieves consistent and reliable data quality evaluation under different construction benchmark dataset sizes, highlighting its robustness and adaptability. Collectively, these contributions lay the groundwork for more data-efficient and effective computer vision model deployment in dynamic and complex construction environments.

2 Data Quality Evaluation in the AI and Construction Domains

While extensive efforts have focused on enlarging datasets to enhance computer vision model performance, such quantity-oriented strategies often overlook the quality of individual samples, leading to redundant or noisy data that undermines model efficacy and efficiency. Consequently, there is a pressing need to shift focus from mere volume to data quality. However, unlike evaluating an entire dataset—which is straightforward via aggregate metrics—assessing the actual contribution of a single training sample remains a complex challenge.

One foundational strategy for isolating the specific contribution of a training sample is the Leave-One-Out (LOO) method. Conceptually, LOO operates on a perturbation principle: it quantifies the value of a datum by excluding it from the training set, retraining the model from scratch, and measuring the resultant shift in test performance. A notable degradation in accuracy signals that the omitted sample contained unique and valuable information. However, since the computational cost of literal retraining for every sample is prohibitive, Koh and Liang [8] addressed this bottleneck by proposing the use of influence functions. By differentiating through the model's training parameters, they derived a closed-form approximation that estimates how upweighting or perturbing a specific point would affect model parameters and loss, all without the need for actual retraining. This technique has proven instrumental for diagnostic applications, such as debugging domain mismatches, explaining model behavior, and detecting mislabeled examples. Yet, despite its rigorous theoretical basis, the computational complexity of calculating Hessian-vector products remains a barrier for large-scale datasets and deep architectures [9]. Furthermore, LOO suffers from a structural deficiency known as the "masking effect": in the presence of redundant or correlated data, it tends to assign near-zero value to individual samples because removing a single instance rarely impacts global performance, thus failing to recognize their collective importance [10].

To capture these complex interactions between data points, a parallel line of research leverages the Shapley value, a solution concept originally derived from cooperative game theory [11]. While originally utilized for interpreting supervised models and performing feature attribution, this framework has been systematically extended to the realm of data valuation. Spearheading this adaptation, Ghorbani and Zou [12] modeled the training process as a cooperative game where data points act as players working together to maximize model performance. To overcome the combinatorial explosion inherent in exact Shapley value calculations, they introduced a truncated Monte Carlo sampling strategy (TMC-Shapley). This stochastic

approximation method renders the evaluation computationally feasible by estimating marginal contributions over a subset of permutations rather than the full space. However, even with such algorithmic optimizations, the computational burden remains substantial, particularly when scaling to massive computer vision datasets [13]. Moreover, a critical limitation of both Shapley-based and influence function approaches is their static nature; they typically evaluate sample importance as an offline process completely decoupled from the active training loop. This separation prevents the joint optimization of data valuation and model learning, which may lead to value estimates that are suboptimal for maximizing downstream task performance [9].

Beyond traditional greedy search methods, recent research has explored more adaptive strategies, most notably Active Learning. This paradigm prioritizes selecting the most informative unlabeled samples for annotation, typically guided by uncertainty or entropy metrics. In the construction domain, such techniques have proven effective for minimizing labeling effort; for instance, Kim et al. [14] developed a deep active learning framework that significantly reduced the data requirements for object detection by automatically querying meaningful images. However, active learning relies heavily on heuristics that often lack robustness in complex visual conditions [15]. More critically, its primary objective is to acquire labels for newly collected data rather than to retrospectively identify and remove harmful samples from existing datasets. Consequently, it falls short in unlocking the full potential of models trained on noisy or redundant data [16].

To bridge these gaps, Reinforcement Learning (RL) has emerged as a promising avenue, offering the ability to model fine-grained data valuation through dynamic trial-and-error interactions. Yoon et al. [17] pioneered this direction with Data Valuation using Reinforcement Learning (DVRL), a framework that employs the REINFORCE algorithm to learn probabilistic weights for training samples. Unlike LOO and Shapley value, DVRL's computational complexity does not scale directly with dataset size, making it highly scalable for large-scale applications. Building on this, Batanero et al. [9] introduced RLBoost, a trajectory-free approach leveraging Proximal Policy Optimization (PPO). By integrating the valuation process directly into the supervised training loop, RLBoost achieves superior evaluation performance compared to both traditional methods and DVRL, demonstrating the potential of RL for joint data-model optimization.

Despite these advancements, a critical methodological gap remains: existing RL-based data quality evaluation methods [9,17] are predominantly limited to simpler tasks such as classification or regression. Extending these frameworks to object detection is not a trivial adaptation; unlike prior methods that rely on simple classification accuracy, it requires redefining the RL reward formulation and state representations to handle significantly higher-dimensional outputs, intricate architectures, and heightened sensitivity to visual noise. This challenge is particularly acute in the construction domain, where data is inherently diverse and prone to redundancy. Moreover, the robustness of these methods under varying dataset scales and diverse visual contexts has not been rigorously examined, leaving their scalability and generalization capabilities unverified. To bridge these gaps, we propose an RL-powered data quality evaluation and sampling framework explicitly designed for object detection within construction environments. We systematically investigate its effectiveness on a representative benchmark dataset of differing sizes and characteristics. Uniquely, our approach involves the joint optimization of an RL-evaluator and a visual detector within a unified RL environment. We then conduct controlled data curation experiments to benchmark the detection performance of models trained on these selectively curated data subsets.

3 RL-powered Data Quality Evaluation Framework

3.1 Problem Definition

The overall workflow of the proposed framework is illustrated in Figure 1. It integrates two learnable components: (i) an object detector $f(;,\omega)$ and (ii) the RL-evaluator $h(;,\theta)$, trained within a unified environment. The RL-evaluator interacts with the detector by a closed feedback loop, generating quality scores for training samples, selecting subsets for model updates, and refining its policy based on the reward signal derived from the model's validation performance. Through this iterative feedback, the RL-evaluator gradually refines its scoring policy to emphasize informative samples and down-weight noisy or harmful ones, thereby enhancing the detector's performance.

Formally, we denote the training dataset as $D = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is an input image and y_i is its corresponding annotation set. Each annotation consists of n_i object instances, i.e., $y_i = \{b_i^{(1)}, b_i^{(2)}, \dots, b_i^{(n_i)}\}$. Each object instance $b_i^{(j)}$ is represented as a 5-dimensional vector (c, c_x, c_y, w, h), where $c \in \{1, \dots, C\}$ is the class ID and $(c_x, c_y, w, h) \in [0,1]^4$ are the normalized bounding-box coordinates. Model performance is finally evaluated on a test dataset $D^t = \{(x_i^t, y_i^t)\}_{i=1}^{M}$, while a validation dataset $D^v = \{(x_i^v, y_i^v)\}_{i=1}^{L}$, assumed to share the same distribution as the test set, is used both to monitor performance during training and to provide

feedback to the RL-evaluator.

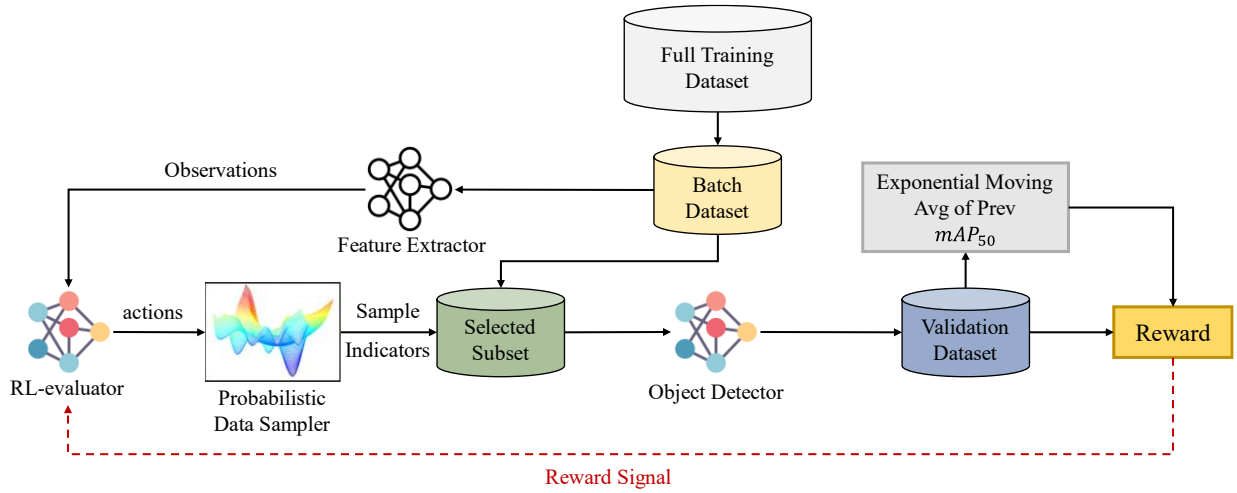

Figure 1. The overall workflow of the RL-powered data quality evaluation framework

In each training epoch, a mini batch $D^b = \{(x_i^b, y_i^b)\}_{i=1}^{B_s} \subset D$ is randomly sampled and passed to the RL-evaluator. The RL-evaluator generates quality score (i.e., selection probability) for each image, estimating its contribution to improving detection performance. A subset $S = \{(x_i^s, y_i^s)\}_{i=1}^{n} \subset D^b$ is then drawn according to these probabilities and used to update the object detector $f(;,\omega)$. The updated model is evaluated on the validation dataset D^v, and the performance is converted into a reward signal that guides the update of the evaluator's policy $h(;,\theta)$.

3.2 RL Formulation

The data quality evaluation can be formulated as Markov Decision Process (MDP), where the RL-evaluator interacts with an environment comprising the detector $f(;,\omega)$, the training data, and the reward derivation process. A standard MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, T)$ involves observing a state $s \in \mathcal{S}$, taking an action $a \in \mathcal{A}$ according to the policy $\pi_\theta(a|s)$, and receiving a reward signal $\mathcal{R}$. In this study, we adopt a contextual bandit setting [18], where the evaluator assigns selection probabilities to samples within each mini-batch and receives immediate rewards from validation performance. As a result, the state transition $\mathcal{P}$, time horizon T, and discount factor γ are not relevant and are excluded from the formulation. In the following, we detail the definitions of state, action, and reward under this bandit MDP setting.

3.2.1 State

Each sample state s is its image-level feature embedding extracted via a ResNet50 [19] pretrained on ImageNet. Using features instead of raw pixels reduces input dimensionality, improves efficiency, and provides semantically rich, task-agnostic representations that enhance generalizability.

3.2.2 Action

For a sample (x_i, y_i) with state s_i, the evaluator outputs a probability $p_i = h_\theta(s_i) \in [0,1]$, which is then passed to a probabilistic data sampler characterized by a Bernoulli distribution to decide selection z_i : $z_i \sim Bernoulli(p_i)$, where $z_i \in \{0,1\}$. Here, $z_i = 1$ means the sample is chosen for training the detector, while $z_i = 0$ excludes it. Since this action directly determines the sampling behavior, the probability p_i serves as a proxy for data quality.

3.2.3 Reward

The reward signal measures the benefit of selected samples by evaluating detection accuracy on the validation set. We use mAP_{50} as the performance metric, and define the reward at iteration t as the difference between current mAP_{50} and the exponential moving average (EMA) of past values:

$$\mathcal{R}_t = mAP_{50_t} - EMA_{t-1} \quad (1)$$

$$EMA_t = \alpha \cdot mAP_{50_t} + (1-\alpha) \cdot EMA_{t-1} \quad (2)$$

where $\alpha \in (0,1)$ is the smoothing parameter for EMA.

3.3 Network Structures: Object Detector and RL-evaluator

The detector $f(;,\omega)$ can be any learnable object detection architecture. Specifically, $f(;,\omega): x_i^s \rightarrow y_i^s$ is

trained by minimizing the detection loss $\mathcal{L}_{det}$ on the selected subset S:

$$\omega \leftarrow \underset{\omega}{argmin} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{det}(f(x_i^s; \omega), y_i^s) \quad (3)$$

The RL-evaluator $h(;,\theta)$ is optimized using the PPO algorithm [20], which generally outperforms standard Policy Gradient methods such as REINFORCE. PPO offers two key advantages: (i) a clipping mechanism to prevent excessively large policy updates and (ii) advantage estimation to evaluate the relative benefit of actions. The evaluator follows an actor–critic paradigm: the actor $\pi_\theta(a|s)$ models the sampling policy, while the critic $V_\emptyset(s)$ estimates the expected return, yielding the advantage $A = \mathcal{R} - V_\emptyset(s)$. The actor is trained using the clipped surrogate loss:

$$\mathcal{L}^{clip}(\theta) = \mathbb{E}_{(s,a)}[min(r_\theta(a|s)A, clip(r_\theta(a|s), 1 - \epsilon, 1 + \epsilon)A)] \quad (4)$$

where $r_\theta = \frac{\pi_\theta(a|s)}{\pi_{\theta old}(a|s)}$ is the probability ratio between new and old policies, and ϵ is a clipping hyperparameter that bounds the update within $[1 - \epsilon, 1 + \epsilon]$.

The critic network is optimized by minimizing the squared value loss:

$$\mathcal{L}^v(\emptyset) = \|\mathcal{R} - V_\emptyset(s)\|^2 \quad (5)$$

Additionally, an entropy regularization term is introduced to encourage exploration:

$$S(\theta) = -\pi_\theta(a|s) log \pi_\theta(a|s) \quad (6)$$

The final PPO objective integrates all three components into a joint loss:

$$\mathcal{L}_{PPO}(\emptyset, \theta) = \mathcal{L}^{clip}(\theta) - c_1 \mathcal{L}^v(\emptyset) + c_2 S(\theta) \quad (7)$$

where c_1 and c_2 are hyperparameters that balance the critic and entropy terms.

This formulation enables joint optimization of the detector and RL-evaluator within the proposed framework, ensuring tight coupling between data selection and task performance.

4 Experiment, Results and Discussion

We conducted an experiment to address three knowledge gaps: (i) whether removing low-quality images can improve model performance compared to using the full dataset; (ii) whether RL-powered evaluation can extend to complex tasks like object detection in construction; and (iii) how robust such methods are under varying dataset sizes.

We focused on detecting human workers, given their prevalence and importance in construction safety and productivity monitoring. Experiments were performed on the MOCS dataset [7], a publicly available benchmark which contains worker annotations collected from diverse construction sites and is well suited for evaluating robustness under varying conditions. We first filtered images containing worker instances and constructed three subsets of different sizes (500, 5,000, and 10,000 instances) to represent small, medium, and large dataset sizes. Each subset was split into training and validation sets at a ratio of 8:2. In addition, an independent test set of 5,000 worker instances was built for consistent evaluation.

YOLO11 [21] was used as the detector, chosen for its balance of accuracy and efficiency. Batch sizes were scaled with dataset size—4, 8, and 16 images for the small (500 instances), medium (5,000 instances), and large (10,000 instances) sizes. Each epoch involved 30 inner updates to provide sufficient learning while ensuring timely feedback to the RL-evaluator. Default hyperparameters were used with a learning rate of 0.01.

The RL-evaluator was trained with PPO using a two-layer actor–critic network (64 units, ReLU activations). The actor's output was initialized at mean 0.5 with standard deviation $log(0.2)$. PPO used Adam with a learning rate of 1×10^{-4}, clipping ratio $\epsilon = 0.1$, and an EMA smoothing factor $\alpha = 0.05$. Regarding hyperparameter sensitivity, empirical tests indicated that a tight clipping ratio and a low EMA factor were crucial for maintaining stable convergence. Mini batch sizes B_s were 100, 200, and 400 for the small, medium, and large datasets, respectively. PPO training was run for 1,000 epochs, while all other hyperparameters followed standard defaults. The joint training of the RL-evaluator required approximately 48 hours on a single NVIDIA RTX 4090 GPU. To account for the stochastic nature of RL training, experiments were conducted across three independent runs for each dataset size.

The RL-evaluator's output probabilities $p_i = h_\theta(s_i) \in [0,1]$ were used as image quality scores. After sorting image qualities based on these scores, four data utilization strategies were compared:

1. Full data: training on the entire dataset as a baseline.
2. Low-quality removal: progressively removing the lowest-scoring 5%, 10%, … of samples.
3. High-quality removal: removing the top-scoring samples at the same rates.
4. Random removal: removing the same proportions randomly.

Model performance was measured by average precision (AP) at IoU 0.5 (AP_{50}) and IoU 0.5–0.95 (AP_{50-95}), reflecting both detection accuracy and localization quality.

We first report the validation results on the MOCS dataset to verify whether the RL-evaluator can reliably distinguish data quality during training. As shown in

Figure 2, removing a small proportion of the lowest-quality samples consistently improved performance compared to training with the full dataset, whereas removing random samples led to gradual performance degradation. Moreover, progressively removing high-quality samples resulted in the greatest performance reduction. These trends confirm that the RL-evaluator learns an effective policy for differentiating useful from harmful samples.

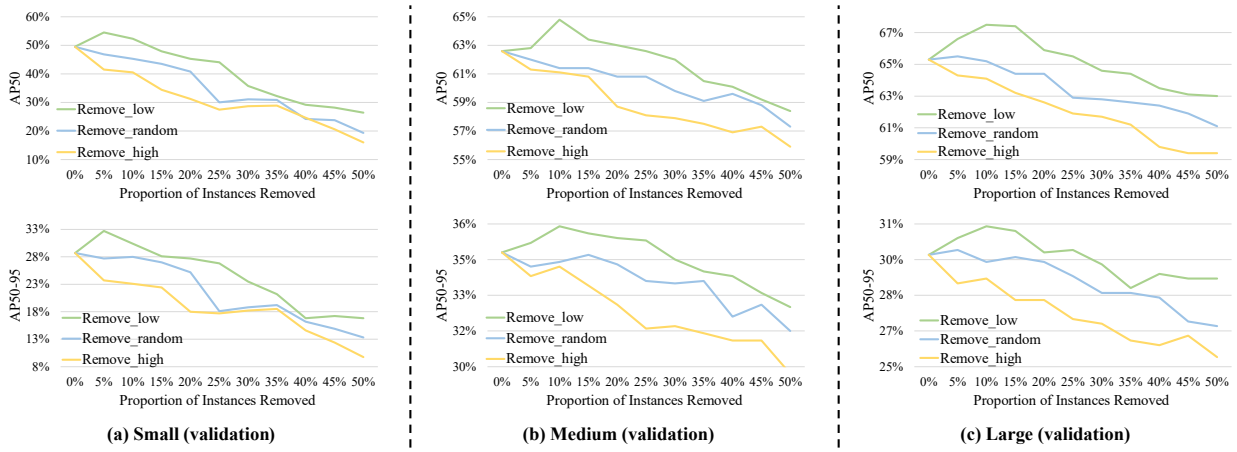

Figure 2. Validation results on MOCS dataset across different dataset sizes

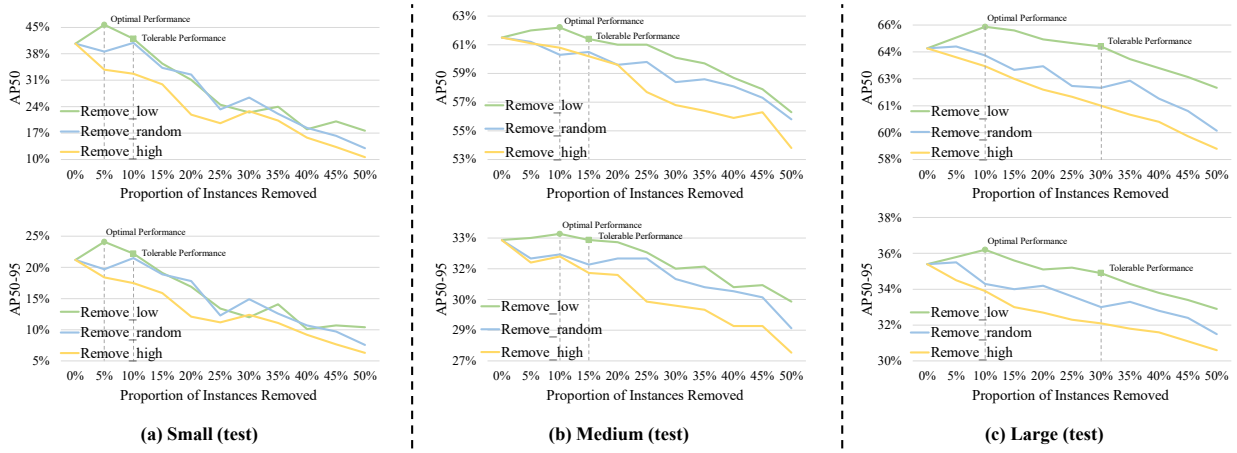

Figure 3. Test results on MOCS dataset across different dataset sizes

We then evaluate our framework on the test dataset. As summarized in Figure 3, the test outcomes are consistent with the validation results. For example, in the small dataset setting, training on the full dataset achieved an AP_{50} of 40.7%. After removing 5% of the lowest-quality samples identified by the RL-evaluator, detector performance improved to 45.7%. In contrast, random removal of the same proportion reduced accuracy to 38.6%, while discarding the top 5% of high-quality samples led to the most severe drop (–6.9%). Similar patterns were also evident in the medium and large dataset settings. Moreover, the benefits were not limited to AP_{50} along: consistent gains were also observed in AP_{50-95}, indicating that the learned data quality scores capture both overall accuracy and localization precision, showing the framework's reliability even with a coarse reward metric. These findings first confirm the conventional belief that "more data leads to better performance," as randomly discarding samples consistently degraded model accuracy. However, this principle does not hold universally. The results validate our alternative hypothesis: selectively removing low-quality samples can in fact yield higher performance than using the full dataset. This counterintuitive outcome underscores the heterogeneous nature of real-world training data—where mislabeled, redundant, or noisy samples can obscure gradient signals and impede learning. By removing such harmful data, the training process becomes clearer and more focused, enabling the detection model to leverage high-quality data more

effectively. Conversely, discarding high-quality samples deprives the model of essential visual and contextual information, resulting in sharp performance degradation.

We further observe that the proportion of removable low-quality samples varies with dataset size. In general, both the optimal removal rate—at which performance peaks—and the tolerable removal rate—at which performance remains comparable to the full dataset—increase as dataset scale grows. For example, on the large MOCS dataset, training with the full data yielded 64.2% AP_{50} , while discarding 10% of the lowest-quality samples achieved optimal performance to 65.4%. Moreover, performance remained close to full data baseline even after removing up to 30% of such samples. By contrast, the optimal removal rates for the small and medium sets were 5% and 10%, with tolerable rates of 10% and 15%, respectively. These results highlight a consistent trend: larger datasets allow more aggressive pruning without degrading performance, demonstrating that our framework improves data utilization efficiency by enabling superior accuracy with fewer, higher-quality samples.

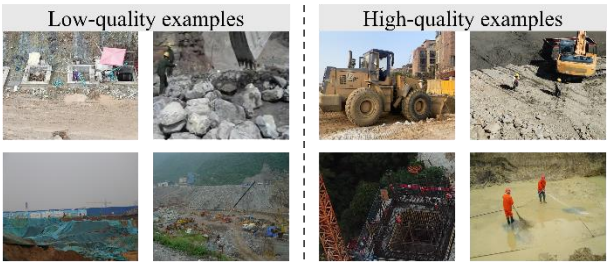

Figure 4. Visual examples of low- and high-quality images in MOCS

Figure 4 shows examples of low- and high-quality images identified by the RL-evaluator on MOCS. It can be observed that low-quality samples often suffered from blur, low resolution, or small and occluded worker instances that provided limited learning signals. In contrast, high-quality samples generally contained clearly visible workers, meaningful context, and diverse construction scenes representative of real conditions. Nevertheless, some exceptions were also observed—certain visually acceptable images contributed little to model performance due to high information redundancy, while some visually challenging ones (e.g., with blur or occlusion) were still valuable as informative hard examples that improve model generalization. These results suggest that the RL-evaluator captures not only superficial visual clarity but also deeper aspects driving the learning dynamics—such as semantic relevance, contextual informativeness (e.g., rare poses or diverse backgrounds), and annotation reliability, offering a more reliable alternative to heuristic or manual filtering.

In summary, the proposed RL-powered framework demonstrates strong potential for improving computer vision in construction by enhancing both detection accuracy and data efficiency. Across different dataset scales, it consistently boosted performance by identifying and removing low-quality samples, validating our alternative hypothesis that low-quality data removing can outperform training on the full dataset. Moreover, the framework enables more efficient training data utilization. By discarding a substantial amount of low-quality data, the model still achieved equal or even superior accuracy compared to using the entire dataset.

5 Conclusion and Future Work

This study presents an RL-powered data quality evaluation framework for computer vision in construction. Experiments on the MOCS dataset across different scales demonstrated that removing a small proportion of the lowest-quality samples—identified by the RL-evaluator—consistently outperformed training on the entire dataset. Furthermore, even after discarding a substantial amount of low-quality data, the detector still achieved competent or even better performance relative to training on the full dataset, thereby significantly reducing computational and data storage overhead. These findings challenge the common belief that more training data leads to better performance and instead highlight the crucial role of data quality and optimizing data utilization to enhancing computer vision model training.

Looking ahead, several research directions remain open. Firstly, our work focused on single-class worker detection; extending the framework to multi-class detection and more complex tasks such as segmentation or pose estimation would help assess its generalizability. Moreover, efforts could also validate the robustness of RL-powered data quality evaluation and broaden its applicability beyond construction vision to other data-intensive AI domains.

6 Acknowledgements

This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (RS-2026-25524415).

References

[1] R. Cai, Z. Guo, X. Chen, J. Li, Y. Tan, J. Tang, Automatic identification of integrated construction elements using open-set object detection based on image and text modality fusion, Advanced Engineering Informatics 64 (2025) 103075. https://doi.org/10.1016/j.aei.2024.103075.

[2] C. Chen, B. Xiao, Y. Zhang, Z. Zhu, Automatic vision-based calculation of excavator earthmoving productivity using zero-shot learning activity recognition, Automation in Construction 146 (2023) 104702. https://doi.org/10.1016/j.autcon.2022.104702.

[3] M. Park, A.S. Kulinan, D.Q. Tran, J. Bak, S. Park, Preventing falls from floor openings using quadrilateral detection and construction worker pose-estimation, Automation in Construction 165 (2024) 105536. https://doi.org/10.1016/j.autcon.2024.105536.

[4] A.S. Kulinan, M. Park, P.P.W. Aung, G. Cha, S. Park, Advancing construction site workforce safety monitoring through BIM and computer vision integration, Automation in Construction 158 (2024) 105227. https://doi.org/10.1016/j.autcon.2023.105227.

[5] B. Ekanayake, J.K.W. Wong, A.A.F. Fini, P. Smith, V. Thengane, Deep learning-based computer vision in project management: Automating indoor construction progress monitoring, Project Leadership and Society 5 (2024) 100149. https://doi.org/10.1016/j.plas.2024.100149.

[6] J. Kim, J. Hwang, I. Jeong, S. Chi, J. Seo, J. Kim, Generalized vision-based framework for construction productivity analysis using a standard classification system, Automation in Construction 165 (2024) 105504. https://doi.org/10.1016/j.autcon.2024.105504.

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

[8] P.W. Koh, P. Liang, Understanding Black-box Predictions via Influence Functions, (2020). https://doi.org/10.48550/arXiv.1703.04730.

[9] E.A. Batanero, Á.F. Pascual, Á.B. Jiménez, RLBoost: Boosting supervised models using deep reinforcement learning, Neurocomputing 618 (2025) 128815. https://doi.org/10.1016/j.neucom.2024.128815.

[10] Y. Kwon, J. Zou, Beta Shapley: a Unified and Noise-reduced Data Valuation Framework for Machine Learning, (2022). https://doi.org/10.48550/arXiv.2110.14049.

[11] L.S. Shapley, A value for n-person games, (1953). https://www.torrossa.com/gs/resourceProxy?an=5641636&publisher=FZO137#page=87 (accessed August 14, 2025).

[12] A. Ghorbani, J. Zou, Data Shapley: Equitable Valuation of Data for Machine Learning, in: Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019: pp. 2242–2251. https://proceedings.mlr.press/v97/ghorbani19c.html (accessed August 14, 2025).

[13] R. Jia, D. Dao, B. Wang, F.A. Hubis, N. Hynes, N.M. Gürel, B. Li, C. Zhang, D. Song, C.J. Spanos, Towards Efficient Data Valuation Based on the Shapley Value, in: Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, PMLR, 2019: pp. 1167–1176. https://proceedings.mlr.press/v89/jia19a.html (accessed September 3, 2025).

[14] J. Kim, J. Hwang, S. Chi, J. Seo, Towards database-free vision-based monitoring on construction sites: A deep active learning approach, Automation in Construction 120 (2020) 103376. https://doi.org/10.1016/j.autcon.2020.103376.

[15] B. Settles, Active Learning Literature Survey, University of Wisconsin-Madison Department of Computer Sciences, 2009. https://minds.wisconsin.edu/handle/1793/60660 (accessed September 3, 2025).

[16] A. Nguyen, J. Yosinski, J. Clune, Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images, in: 2015: pp. 427–436. https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Nguyen_Deep_Neural_Networks_2015_CVPR_paper.html (accessed September 3, 2025).

[17] J. Yoon, S. Arik, T. Pfister, Data Valuation using Reinforcement Learning, in: Proceedings of the 37th International Conference on Machine Learning, PMLR, 2020: pp. 10842–10851. https://proceedings.mlr.press/v119/yoon20a.html (accessed August 16, 2025).

[18] G. Neu, A. Antos, A. György, C. Szepesvári, Online Markov Decision Processes under Bandit Feedback, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2010. https://proceedings.neurips.cc/paper/2010/hash/7bb060764a818184ebb1cc0d43d382aa-Abstract.html (accessed September 4, 2025).

[19] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016: pp. 770–778. https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html (accessed August 27, 2025).

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, (2017). https://doi.org/10.48550/arXiv.1707.06347.

[21] R. Khanam, M. Hussain, YOLOv11: An Overview of the Key Architectural Enhancements, (2024). https://doi.org/10.48550/arXiv.2410.17725.