

Tracking-Informed Missed Detection Mining for Effective MLOps: An Industrial Case Study on Warehouse Worker Detection

Kazuma Kano
Nagoya University
Nagoya, Japan
kazuma@ucl.nuee.nagoya-u.ac.jp

Yuki Mori
Nagoya University
Nagoya, Japan
ymori@ucl.nuee.nagoya-u.ac.jp

Kenta Urano
Nagoya University
Nagoya, Japan
urano@nagoya-u.jp

Takuro Yonezawa
Nagoya University
Nagoya, Japan
takuro@nagoya-u.jp

Nobuo Kawaguchi
Nagoya University
Nagoya, Japan
kawaguti@nagoya-u.jp

Abstract

Visual object detection offers potential to enable various applications, including work analysis for higher productivity in industrial sites. However, particularly in continuous system operation, the detection accuracy can drop when dynamic environments introduce unseen conditions. In this case, continual and iterative retraining is essential for sustaining and enhancing the models' performance. To facilitate effective retraining cycles, we designed MiMiOps, a Machine Learning Operations (MLOps) framework that integrates missed detection mining via tracking. This framework actively remedies model vulnerabilities by prioritizing retraining on hard samples. It also mitigates annotation costs by automatically interpolating the bounding boxes. We built a practical system and applied it to worker detection in a logistics warehouse. An evaluation using actual footage demonstrated the improved detection accuracy through the retraining cycles.

CCS Concepts

• **Computing methodologies** → **Object detection; Tracking;** • **Theory of computation** → **Active learning.**

Keywords

Annotation automation, Fixed camera, Hard example mining, Human in the loop, Pseudo-labeling

ACM Reference Format:

Kazuma Kano, Yuki Mori, Kenta Urano, Takuro Yonezawa, and Nobuo Kawaguchi. 2026. Tracking-Informed Missed Detection Mining for Effective MLOps: An Industrial Case Study on Warehouse Worker Detection. In *Companion of the 2026 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Companion '26)*, October 11–15, 2026, Shanghai, China. ACM, New York, NY, USA, 7 pages. <https://doi.org/XXXXXXX.XXXXXXX>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UbiComp Companion '26, Shanghai, China

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2018/06
<https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

Object detection with cameras plays an active role in various fields such as security and healthcare. It also shows potential for work analysis to enhance productivity and work quality in industrial sites [4, 13]. Modern object detection technologies are based on deep learning and achieve high accuracy. Nevertheless, general-purpose models often struggle to handle top-down views, wide-angle lens distortion, and other specialized industrial settings. This limitation necessitates fine-tuning or domain adaptation. Moreover, for the continuous operation of object detection systems, sporadic events and temporal changes may matter. For example, moving packages sometimes occlude the detection targets heavily in logistics warehouses. Seasonal changes in clothing and lens dirt can also lead to the domain shift problem. Since it is infeasible to create a model that covers all such possible situations in advance, continual and iterative model updates are crucial for sustaining and enhancing its performance. Therefore, we focus on Machine Learning Operations (MLOps), which streamlines the entire process, including model retraining and validation [28].

A major practical challenge in training object detection models lies in the human effort for annotation. Because models tend to perform poorly on exceptional cases rather than typical ones, merely increasing the amount of training data yields limited improvement in accuracy. Here, active learning, which prioritizes annotation for highly informative samples, is a cost-effective solution [29]. We propose MiMiOps incorporating active learning into MLOps of object detection models, as illustrated in Figure 1. This framework automatically extracts frames containing missed detections and interpolates their bounding boxes based on tracking results. Then, it adds the frame images to the ground-truth dataset after manually revising the bounding boxes. This process thereby emphasizes retraining on samples where the model formerly failed to detect. In addition, it allows creating annotated data with less effort compared to fully manual annotation. We implemented a system with features such as a web-based user interface, anticipating the involvement of non-technical personnel. Subsequently, we applied it to worker detection in a logistics warehouse and conducted 14 retraining cycles using over 150 hours of actual video footage in total. The result showed improvement in detection accuracy through the cycles, driven by a particular reduction in missed detections. This study aims to present a hands-on reference for MLOps of object detection models in the wild.

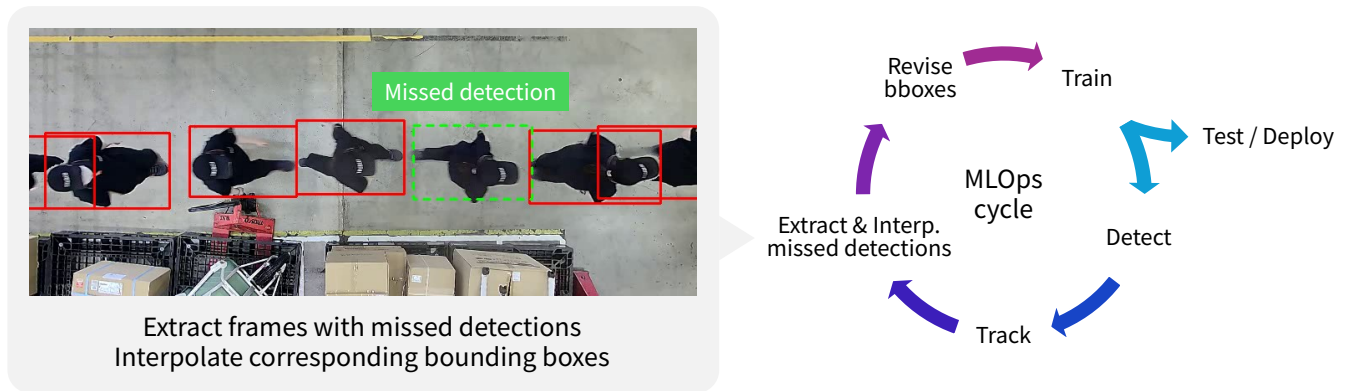


Figure 1: Training and Annotation Cycle of MiMiOps Framework.

2 Related Work

2.1 MLOps

MLOps consistently assists machine learning lifecycles from model training and versioning to data collection and annotation [28]. It is especially helpful for continuous system operation. Kukkaro et al. verified the effectiveness of continual learning on fog devices in a simulated factory environment [14]. Nevertheless, most existing examples of MLOps practice have been driven by industrial companies, and research reports with actual system construction remain relatively scarce. Although open-source tools for MLOps, such as MLflow [2] and Kubeflow [20], have been developed, building in-house systems can be more appropriate depending on the project scale and integration with other systems.

2.2 Self-Supervised Representation Learning and Transfer Learning

Pretraining model backbones on a large volume of unlabeled data is a common approach to achieve high generalization performance with limited labeled data [30]. He et al. revealed that Masked Autoencoders (MAE) can acquire powerful visual representations [6]. Siméoni et al. released weights of DINOv3 trained on over 1.6 billion images [26]. However, applying these methods to downstream tasks such as object detection still requires annotated ground truth. Furthermore, the effectiveness of publicly available foundation models is limited in specialized situations like industrial sites.

2.3 Partial Automation of Annotation

Many studies have attempted to reduce human effort for annotation in various ways. An approach that utilizes predictions of one model as labels for another is called pseudo-labeling [11]. Chen et al. proposed Point-DETR, which predicts pseudo-bounding boxes from single-point annotations for each object [3]. Li et al. proposed SIOD, which assumes annotations of only one instance per class per image and assigns pseudo-labels to the remaining instances based on their similarity [15]. These methods fall into weakly supervised learning, allowing for model training with coarse ground truth. Yet pseudo-labeling relying on visual features is likely to be inaccurate when clues are scarce due to occlusion or when the same-class ob-

jects exhibit appearance variations, such as people walking versus crouching.

Some studies have exploited temporal information in videos to facilitate image annotation. Mahmud et al. proposed SSVOD, which predicts bounding boxes across multiple consecutive frames using a teacher model and adopts only those consistent with optical flow results as pseudo-labels for the student [18]. However, since it updates the teacher weights as an exponential moving average of the student, erroneous teacher predictions can cause the student to learn from incorrect data, leading to performance degradation in the teacher itself. This vicious circle is known as the confirmation bias issue. Sivapuram et al. proposed SA-LfV, which conducts single-object tracking on an annotated object in a given frame and uses its bounding boxes from frames with high tracking confidence as pseudo-labels [27]. However, it does not account for situations where multiple targets appear simultaneously.

We previously developed a semi-automated annotation framework that extracts moving objects with optical flow and groups their embedded representations through clustering [7, 12]. It enables the generation of reliable ground truth with minimal effort by setting bounding boxes automatically and assigning labels to multiple objects collectively. Yet it still has the limitation that the generated data does not contain stationary objects. In fact, when we trained a detection model of warehouse workers on a dataset constructed using this framework, the model tended to miss workers remaining in place for inspection.

2.4 Active Learning

Active learning reduces the amount of labeled data required for model training by automatically selecting samples to be annotated [29]. Park et al. introduced an additional head to estimate the confidence of object detection models and calculated data difficulty based on the composition of bounding boxes [21]. Yang et al. proposed PPAL, which quantifies both uncertainty and diversity of data regardless of the structure of object detection models [32]. Alternatively, Lin et al. [16] and Sajid et al. [25] addressed retraining of object detection models on missed or low-confidence bounding boxes, similar to this study. One of the primary differences is that we leave room for human intervention to prevent confirmation bias while keeping human workloads to a practical minimum.

3 Design and Implementation

3.1 Framework Design

To efficiently maintain and enhance the performance of visual object detection models, we propose MiMiOps, an MLOps framework based on Missed detection Mining via tracking. It aims to reduce human effort by partial annotation automation while improving training effectiveness by active learning. This framework can be applied to arbitrary detection models in arbitrary environments as long as the target objects are trackable. It is also compatible with transfer learning from pretrained models. The flow of a cycle from model training to ground truth creation is outlined below. We will describe the details of each step in what follows.

- (1) Split the ground-truth dataset for training and validation.
- (2) Fine-tune the detection model.
- (3) Conduct detection on every frame of videos.
- (4) Apply tracking to the detected objects across frames.
- (5) Extract frames containing missed detections.
- (6) Interpolate bounding boxes of the missed detections.
- (7) Filter out uncertain or redundant frames.
- (8) Manually review and correct the bounding boxes.
- (9) Add the frame images to the ground-truth dataset.

Step 1. prepares a dataset composed of all annotated images, including those made in prior cycles. Then, randomly splits it into training and validation subsets.

Step 2. fine-tunes the detection model using them. We update the model weights to minimize loss on the training subset and adopt those from the epoch with the best Average Precision (AP) on the validation subset. Note that if the weights are reinitialized at every cycle, the loss will take longer to converge. So, we resume training by inheriting the weights from the previous cycle.

Step 3. uses the fine-tuned model to detect target objects in each frame of the given videos. It is desirable to choose videos captured at different times from prior cycles to enrich data patterns.

Step 4. tracks the target objects throughout all frames in the videos. We employ a multi-object tracking method based on the Tracking-by-Detection (TBD) paradigm, which chains corresponding bounding boxes detected beforehand across frames [5]. As a result, each object trajectory will be expressed as a sequence of the bounding boxes at frames where the object was detected. That is, if no bounding box is associated with the trajectory at some intermediate frames, it implies that detection may have failed at those frames. In this paper, we customized ByteTrack [34], a representative TBD-based method, for multiple wide-angle cameras [19]. We performed tracking in a global coordinate system and projected the resulting trajectories onto the local coordinate systems of each camera.

Step 5. extracts frames that contain missed detections based on the tracking results. Let I_{det}^τ denote the set of frame indices where bounding boxes associated with a trajectory τ exist. Let $\mathcal{T}$ denote the set of all trajectories. Then, the equation below yields the set I_{miss} of frame indices with missed detections.

$$I_{miss} = \bigcup_{\tau \in \mathcal{T}} \{i \in \mathbb{N} \mid \min I_{det}^\tau < i < \max I_{det}^\tau \wedge i \notin I_{det}^\tau\} \quad (1)$$

Step 6. linearly interpolates the bounding boxes of the missed detections based on the detection and tracking results. Specifically, the bounding box in a frame not detected is estimated by internally dividing the ones in the last frame detected before it and the first frame detected after it, as depicted in Figure 2. Let $(x^{\tau,l}, y^{\tau,l})$ denote the top-left coordinates of a bounding box associated with a trajectory τ in the l -th frame. Then, the equations below give the interpolated top-left coordinates $(\tilde{x}^{\tau,m}, \tilde{y}^{\tau,m})$ in the m -th frame. The same holds for the top-right, bottom-left, and bottom-right coordinates as well.

$$l = \max \{i \in \mathbb{N} \mid i < m \wedge i \in I_{det}^\tau\} \quad (2)$$

$$n = \min \{i \in \mathbb{N} \mid i > m \wedge i \in I_{det}^\tau\} \quad (3)$$

$$\tilde{x}^{\tau,m} = \frac{(n-m)x^{\tau,l} + (m-l)x^{\tau,n}}{n-l} \quad (4)$$

$$\tilde{y}^{\tau,m} = \frac{(n-m)y^{\tau,l} + (m-l)y^{\tau,n}}{n-l} \quad (5)$$

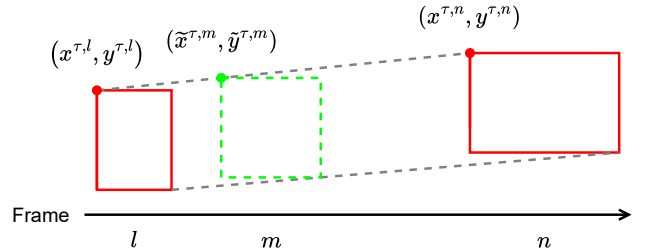


Figure 2: Linear Interpolation of Missed Detection.

Step 7. selects which frames to add to the ground truth among frames with missed detections based on two criteria: uncertainty and similarity. First, when an object remains undetected for an extended duration, its interpolated bounding boxes may deviate substantially from its actual positions. To prevent degradation in data quality and annotation efficiency, we exclude frames that contain interpolated bounding boxes undetected for a period longer than T_{miss} . Second, frames temporally close to each other tend to carry overlapping information. To improve training efficiency, we also skip frames within a period T_{near} from an already adopted frame.

Step 8. involves humans revising the adopted frame images and bounding boxes. During revision, if an interpolated bounding box appreciably shifted away from the actual object or enclosed a non-target region, annotators can correct it by hand. Although this is the only manual work in the cycle, it is easier than fully manual annotation. Additionally, it is less prone to annotation omissions caused by human oversight.

Step 9. adds the annotated frame images to the ground truth. These samples are expected to contain patterns that the current model finds difficult to recognize. This framework augments the ground truth progressively by repeating the above cycle, leading to higher generalization performance.

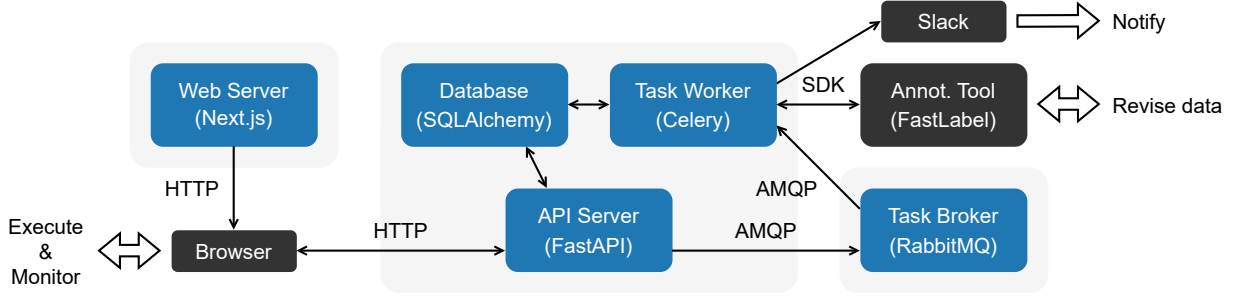


Figure 3: Architecture of MiMiOps System.

3.2 System Implementation

We developed a practical system in accordance with the MiMiOps framework to facilitate execution of the cycles. Figure 3 illustrates the overall architecture. The black nodes represent external interfaces where users interact. The blue nodes represent backends where the system handles internal processing. Each gray block operates as an independent container and communicates with others over the network.

3.2.1 Interface. Primary manipulations related to process execution and monitoring are accessible through a dashboard implemented as a web application. It intends to support collaboration with personnel unfamiliar with technical work such as programming. For reference, Figure 4 shows a dashboard screen for cycle monitoring. Additionally, users get push notifications via Slack when processes finish. For revision of frame images and bounding boxes, we utilize an annotation tool on FastLabel¹, a cloud-based AI development platform, as shown in Figure 5.

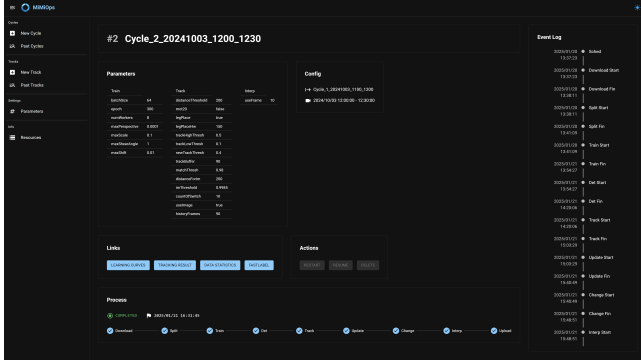


Figure 4: Dashboard Application.

3.2.2 Backend. When the API server receives a request to start a new cycle from the browser, it registers tasks in the queue managed by the broker. Then, the worker sequentially carries out the tasks in the queue and sends a message via Slack bot when completing the final task or producing an error. Here, tasks refer to every automatic process in the cycle, for example, downloading revised data from FastLabel, fine-tuning the model, extracting and interpolating missed detections, and uploading unrevised data to FastLabel.

¹<https://fastlabel.ai/en>

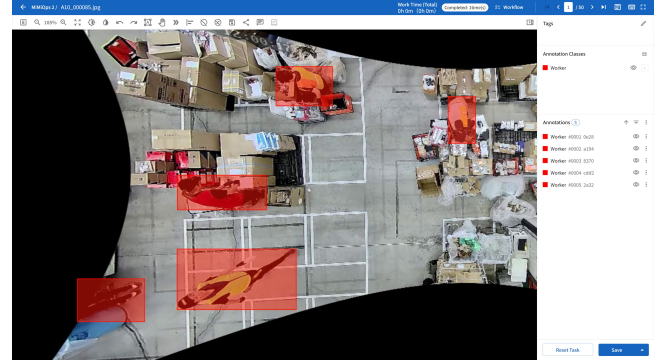


Figure 5: Annotation Tool Provided by FastLabel.

4 Field Evaluation

4.1 Model and Data

We applied the MiMiOps system to a worker detection model in a Japanese logistics warehouse to assess its effectiveness. Wide-angle cameras (H.View HV-800G2A5²) were mounted vertically downward on the ceiling of the inbound and outbound areas, as shown in Figure 6. We employed the extra-large version of YOLOv8 [8] as the detection model. The model had been tuned in advance using ground truth composed of 8515 manually annotated images and 8802 semi-automatically synthesized images [7, 12]. This experiment additionally used video footage from 19 cameras on weekdays. The videos were recorded in full HD resolution at 5 fps and up to 8 Mbps, and were undistorted with Double Sphere camera models [31]. Figure 7 shows an example of the undistorted frame image.

4.2 Execution of MLOps Cycles

We conducted 14 complete cycles from model training to ground truth creation, followed by the training phase into the 15th cycle. The ratio of training to validation subset at fine-tuning was 9 : 1. The training subset was randomly augmented in hue, saturation, brightness, translation, scale, shear, perspective, and flip. The batch size and maximum number of epochs were 64 and 300, respectively. We empirically set both parameters T_{miss} and T_{near} , which control data reliability and diversity, to 2 seconds. This duration accounts

²<https://hviewsmart.com/products/h-view-colorcam-4k-bullet-ai-camera-with-color-night-vision-hv-800g2a5>

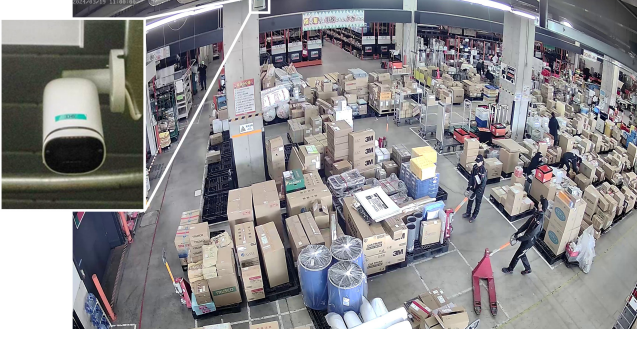


Figure 6: Fixed Camera in Warehouse Inbound Area.

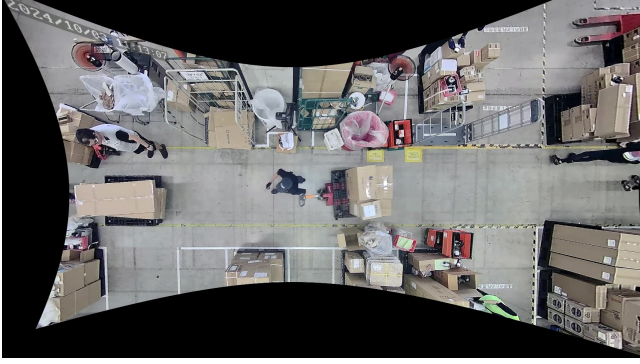


Figure 7: Example of Undistorted Frame Image.

for substantial appearance changes across the associated movements under wide-angle lens distortion. Table 1 summarizes the video durations used and the total ground truth quantities created from them for each cycle. The considerable variation in the ground truth quantity per video duration is seemingly due to differences not only in the accuracy of the detection model but also in the number of detection targets present.

4.3 Creation of Test Data

We manually created test data from 180 minutes of the videos not used in the cycles. At that time, we annotated frames uniformly sampled at 1-minute intervals rather than ones containing missed detections. It intended to ensure fair evaluation regardless of the detection accuracy biases. Note that frames with severe artifacts from transmission errors were discarded. Consequently, the test data comprised 3419 ($\sim 19 \times 180$) images with 10399 annotations.

4.4 Results and Discussions

Table 2 summarizes the AP on the test data for the model at each cycle. Figure 8 presents their transitions across the cycles. Here, the model at the j -th cycle refers to the model fine-tuned in the j -th cycle using the data accumulated up to the $j - 1$ -th cycle. The model at the first cycle is equivalent to the initial state before the cycles. These results indicate a trend of increasing AP as the cycles proceeded. MiMiOps appears to deliver solid improvements in detection accuracy while mitigating manual processes.

Table 1: Video Durations Used and Ground Truth Quantities Created

Cycle	Video Duration (min)	# of Images	# of Annots
1	19×60	3807	19732
2	19×60	1678	7940
3	19×30	1201	5029
4	19×30	660	2069
5	19×30	1271	6166
6	19×30	1241	6152
7	19×30	852	3530
8	19×30	939	4147
9	19×30	2103	11339
10	19×30	900	4783
11	19×30	1162	6799
12	19×30	752	4073
13	19×30	505	2237
14	19×30	487	1414
1–14	19×480	17558	85410

Table 2: Average Precision per Cycle on Test Data

Cycle	AP50:95	AP50	AP75
1	0.508	0.723	0.559
2	0.584	0.840	0.639
3	0.576	0.834	0.632
4	0.594	0.852	0.652
5	0.586	0.835	0.644
6	0.593	0.860	0.655
7	0.594	0.864	0.651
8	0.596	0.867	0.651
9	0.598	0.866	0.661
10	0.636	0.896	0.705
11	0.639	0.905	0.711
12	0.641	0.907	0.712
13	0.637	0.904	0.704
14	0.643	0.907	0.715
15	0.652	0.915	0.722

Nevertheless, compared with AP50 exceeding 0.9, AP50:95 and AP75 seem to leave room for further improvement. A possible factor is misalignment of bounding boxes in the created ground truth. In this study, we interpolated the bounding boxes of missed detections linearly, but they do not always match the actual positions and sizes of the objects. The degree of error correction depends on each annotator; while stricter adjustments could improve data quality, they would also increase annotation effort. Nonlinear interpolation methods, such as spline interpolation, or segmentation techniques, which estimate object boundaries, may help reduce the misalignment. Having said that, the precision of bounding boxes is not very critical in some applications. For example, identifying a pallet inspected by a warehouse worker requires only a positioning resolution of about 1.1 m, the size of a single pallet.

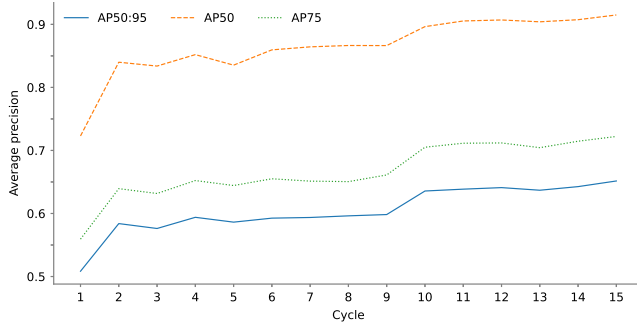


Figure 8: Average Precision per Cycle on Test Data.

Figure 9 plots the precision–recall curves with an Intersection over Union (IoU) threshold of 0.5. Note that it displays only a portion of the region for visibility. This result demonstrates that both recall and precision tended to improve over the cycles. MiMiOps incorporated active learning to suppress missed detections (i.e., false negatives). It likely contributed to the great improvement in recall. On the other hand, the system did not explicitly address false detections (i.e., false positives). The richer patterns in the ground truth seem to have led to more correct detections (i.e., true positives), resulting in an improvement in precision accordingly. The precise extent to which active learning and partial annotation automation contribute to accuracy gains and cost reductions warrants deeper investigation. We leave quantitative analysis to isolate these effects for future work.

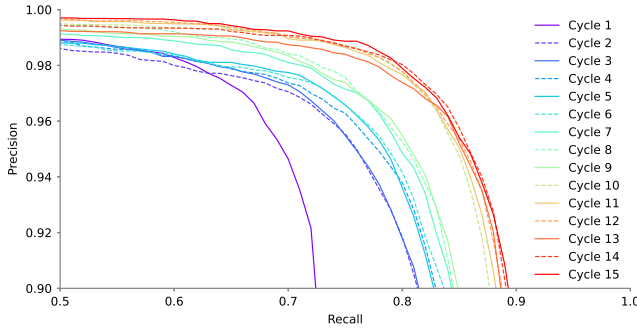


Figure 9: Precision–Recall Curves per Cycle on Test Data.

In this study, we fine-tuned the detection model using all accumulated ground-truth data at every cycle to prevent forgetting knowledge learned in prior cycles. Although this procedure worked sufficiently well in this experiment, it also has limitations. As the cycles progress, the training time and computational amount increase, and the influence of newly added data declines. In the context of continual learning [22], various studies have sought to overcome catastrophic forgetting while preserving adaptability to new data. These approaches include experience replay, which blends informative past samples with new ones [17, 23], and weight regularization, which constrains updates to critical weight parameters [1, 33]. We are considering integrating such techniques into the MiMiOps framework to enable further long-term operation.

5 Model Benchmark

In this paper, we adopted YOLOv8 [8] as the detection model and have engaged in MLOps for nearly 2 years to enhance its accuracy. However, many new models have emerged during this period. Hence, we verify whether alternative models allow the system to extend its capabilities. Specifically, we examine three YOLO models [8–10], each trained with its launch or current version of the Ultralytics library. We also compare RF-DETR [24], a state-of-the-art model from the DETR family. We tuned them using the ground truth accumulated up to the 14th cycle and assessed them by consistent protocol on the test data. Table 3 lists the AP averaged over five trials with different random seeds for each model. The AP exhibited little variation across YOLO versions, whereas it was sensitive to used libraries. As a result, RF-DETR yielded the best scores in our settings, likely attributable in part to its twice as many parameters.

Table 3: Average Precision per Model on Test Data

Model	Ultralytics	Albumentations	AP50:95	AP50	AP75
YOLOv8x	8.0.0	Disabled	0.680	0.930	0.769
YOLOv8x	8.4.70	Enabled	0.672	0.935	0.758
YOLO11x	8.3.0	Disabled	0.660	0.926	0.741
YOLO11x	8.4.70	Enabled	0.673	0.934	0.760
YOLO26x	8.4.0	Disabled	0.646	0.923	0.721
YOLO26x	8.4.70	Enabled	0.674	0.933	0.763
RF-DETR-2XL	—	Enabled	0.698	0.947	0.781
YOLOv8x (Cycle 15)	8.3.52	Disabled	0.652	0.915	0.722

6 Conclusion

In this study, we tackled MLOps for maintaining and enhancing the performance of object detection models efficiently. Our MiMiOps framework improves training effectiveness by active learning, re-training models on frame images containing missed detections. Additionally, it reduces human effort for annotation by utilizing detected or interpolated bounding boxes. We implemented the framework as a practical system and also shared its configuration. Furthermore, we performed the training and annotation cycles for a detection model of warehouse workers. The result confirmed the improvement in AP through the iterations, with a notable increase in recall. At the same time, it also suggested remaining challenges related to bounding box misalignment. Separately, the model benchmark promised further improvement by shifting models. We plan to continue refining and operating this system from now on. We expect this study to serve as a useful reference for building MLOps pipelines of object detection models.

Acknowledgments

This work is partially supported by JSPS KAKENHI (JP22K18422), NEDO (JPNP23003), CSTI SIP3 (JPJ012495), JST BOOST (JPMJBS2422), and SAP SE. We also thank TRUSCO Nakayama Corporation for providing the experimental environment.

References

- [1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory Aware Synapses: Learning What (not) to Forget. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part III*. 144–161. doi:10.1007/978-3-030-01219-9_9
- [2] Andrew Chen, Andy Chow, Aaron Davidson, Arjun DCunha, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Clemens Mewald, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, Avesh Singh, Fen Xie, Matei Zaharia, Richard Zang, Juntai Zheng, and Corey Zumar. 2020. Developments in MLflow: A System to Accelerate the Machine Learning Lifecycle. In *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning (DEEM '20)*. Article 5, 4 pages. doi:10.1145/3399579.3399867
- [3] Liangyu Chen, Tong Yang, Xiangyu Zhang, Wei Zhang, and Jian Sun. 2021. Points as Queries: Weakly Semi-supervised Object Detection by Points. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8819–8828. doi:10.1109/CVPR46437.2021.00871
- [4] Jianqiao Cheng, Connor Verhulst, Pieter De Clercq, Shannon Van De Velde, Steven Sagaert, Marc Mertens, Merwan Birem, Maithili Deshmukh, Neel Broekx, Erwin Rademakers, Abdellatif Bey-Temsamani, and Jean-Edouard Blanquart. 2025. Real-Time Warehouse Monitoring with Ceiling Cameras and Digital Twin for Asset Tracking and Scene Analysis. *Logistics* 9, 4 (2025). doi:10.3390/logistics9040153
- [5] Zhiyu Guan, Zhaofa Wang, Gan Zhang, Luwei Li, Miaomiao Zhang, Zhiping Shi, and Na Jiang. 2025. Multi-object tracking review: retrospective and emerging trend. *Artificial Intelligence Review* 58, 8 (May 2025), 235. doi:10.1007/s10462-025-11212-y
- [6] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked Autoencoders Are Scalable Vision Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 15979–15988. doi:10.1109/CVPR52688.2022.01553
- [7] Keisuke Higashiura, Kodai Yokoyama, Yusuke Asai, Hironori Shimotsato, Kazuma Kano, Shin Katayama, Kenta Urano, Takuro Yonezawa, and Nobuo Kawaguchi. 2024. Semi-Automated Framework for Digitalizing Multi-Product Warehouses with Large Scale Camera Arrays. In *2024 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 98–105. doi:10.1109/PerCom59722.2024.10494498
- [8] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. *Ultralytics YOLOv8*. <https://github.com/ultralytics/ultralytics>
- [9] Glenn Jocher and Jing Qiu. 2024. *Ultralytics YOLO11*. <https://github.com/ultralytics/ultralytics>
- [10] Glenn Jocher, Jing Qiu, Mengyu Liu, Shuai Lyu, Fatih Catay Akyon, and Muhammet Esat Kalfaoglu. 2026. Ultralytics YOLO26: Unified Real-Time End-to-End Vision Models. arXiv:2606.03748 [cs.CV] <https://arxiv.org/abs/2606.03748>
- [11] Patrick Kage, Jay Rothenberger, Pavlos Andreadis, and Dimitrios Diochnos. 2026. A Review of Pseudo-Labeling for Computer Vision. *Journal of Artificial Intelligence Research* 85 (March 2026), 32 pages. doi:10.1613/jair.1.19656
- [12] Kazuma Kano, Yuki Mori, Keisuke Higashiura, Tahera Hossain, Shin Katayama, Kenta Urano, Takuro Yonezawa, and Nobuo Kawaguchi. 2024. Composite Image Generation Using Labeled Segments for Pattern-Rich Dataset without Unannotated Target. In *Companion of the 2024 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '24)*. 507–512. doi:10.1145/3675094.3678447
- [13] Nobuo Kawaguchi, Yusuke Asai, Kazuma Kano, Kairi Takaki, Yuki Mori, Yuma Suzuki, Kisho Watanabe, Yuki Gushi, Shin Katayama, Kenta Urano, Takuro Yonezawa, and Shintaro Hashiguchi. 2025. Digitization Methods for a Logistics Warehouse Towards Digital Twin-Driven Optimization. In *2025 IEEE International Conference on Smart Computing (SMARTCOMP)*. 186–193. doi:10.1109/SMARTCOMP65954.2025.00035
- [14] Ari Kukkaro, Sergio Moreschini, and David Hästbacka. 2024. Continuous Training vs. Transfer Learning on Edge and Fog Environments: A Steam Detection use Case. In *2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 138–141. doi:10.1109/SEAA64295.2024.00029
- [15] Hanjun Li, Xingjia Pan, Ke Yan, Fan Tang, and Wei-Shi Zheng. 2022. SIOD: Single Instance Annotated Per Category Per Image for Object Detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14177–14186. doi:10.1109/CVPR52688.2022.01380
- [16] Yun-Wei Lin, Yuh-Hwan Liu, Yi-Bing Lin, and Jian-Chang Hong. 2023. FenceTalk: Exploring False Negatives in Moving Object Detection. *Algorithms* 16, 10 (2023). doi:10.3390/a16100481
- [17] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. 6470–6479.
- [18] Tanvir Mahmud, Chun-Hao Liu, Burhaneddin Yaman, and Diana Marculescu. 2024. SSVOD: Semi-Supervised Video Object Detection with Sparse Annotations. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 6759–6768. doi:10.1109/WACV57701.2024.00663
- [19] Yuki Mori, Kazuma Kano, Yusuke Asai, Shin Katayama, Kenta Urano, Takuro Yonezawa, and Nobuo Kawaguchi. 2025. Multi-Camera Worker Tracking in Logistics Warehouse Considering Wide-Angle Distortion. arXiv:2510.19432 [cs.CV] <https://arxiv.org/abs/2510.19432>
- [20] Aditya Pandey, Maitreya Sonawane, and Sumit Mamtani. 2022. Deployment of ML Models using Kubeflow on Different Cloud Providers. arXiv:2206.13655 [cs.LG] <https://arxiv.org/abs/2206.13655>
- [21] Younghyun Park, Wonjeong Choi, Soyeong Kim, Dong-Jun Han, and Jaekyun Moon. 2023. Active Learning for Object Detection with Evidential Deep Learning and Hierarchical Uncertainty Aggregation. In *The Eleventh International Conference on Learning Representations*. <https://openreview.net/forum?id=MnEjsw-vj-X>
- [22] Haoxuan Qu, Hossein Rahmani, Li Xu, Bryan Williams, and Jun Liu. 2025. Recent Advances of Continual Learning in Computer Vision: An Overview. *IET Computer Vision* 19, 1 (2025), e70013. doi:10.1049/cvi2.70013
- [23] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. iCaRL: Incremental Classifier and Representation Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5533–5542. doi:10.1109/CVPR.2017.587
- [24] Isaac Robinson, Peter Robicheck, Matvei Popov, Deva Ramanan, and Neehar Peri. 2026. RF-DETR: Neural Architecture Search for Real-Time Detection Transformers. In *The Fourteenth International Conference on Learning Representations*. <https://openreview.net/forum?id=qHm5GePxTh>
- [25] Shoaib Sajid, Zafar Aziz, Odibek Urmonov, and HyungWon Kim. 2024. Improving Object Detection Accuracy with Self-Training Based on Bi-Directional Pseudo Label Recovery. *Electronics* 13, 12 (2024). doi:10.3390/electronics13122230
- [26] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Quab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Santana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. 2026. DINOv3. *Transactions on Machine Learning Research* (2026). <https://openreview.net/forum?id=2NiGyqNjns>
- [27] Arun Kumar Sivapuram, Prashanth Komuravelli, and Rama Krishna Sai Gorthi. 2025. SA-LfV: self-annotated labeling from videos for object detection. *Machine Learning* 114, 1 (Jan. 2025), 24 pages. doi:10.1007/s10994-024-06676-y
- [28] A. I. Ullah Tabassam. 2023. MLOps: A Step Forward to Enterprise Machine Learning. arXiv:2305.19298 [cs.SE] <https://arxiv.org/abs/2305.19298>
- [29] Rinyoichi Takezoe, Xu Liu, Shunan Mao, Marco Tianyu Chen, Zhanpeng Feng, Shiliang Zhang, and Xiaoyu Wang. 2023. Deep Active Learning for Computer Vision: Past and Future. *APSIPA Transactions on Signal and Information Processing* 12, 1 (March 2023), 1–38. doi:10.1561/116.000000057
- [30] Tobias Uelwer, Jan Robine, Stefan Sylvius Wagner, Marc Höftmann, Eric Uppschulte, Sebastian Konietzny, Maike Behrendt, and Stefan Harmeling. 2025. A survey on self-supervised methods for visual representation learning. *Machine Learning* 114, 4 (March 2025), 56 pages. doi:10.1007/s10994-024-06708-7
- [31] Vladyslav Usenko, Nikolaus Demmel, and Daniel Cremers. 2018. The Double Sphere Camera Model. In *2018 International Conference on 3D Vision (3DV)*. 552–560. doi:10.1109/3DV.2018.00069
- [32] Chenhongyi Yang, Lichao Huang, and Elliot J. Crowley. 2024. Plug and Play Active Learning for Object Detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 17784–17793. doi:10.1109/CVPR52733.2024.01684
- [33] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML '17)*. 3987–3995.
- [34] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. 2022. ByteTrack: Multi-object Tracking by Associating Every Detection Box. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*. 1–21. doi:10.1007/978-3-031-20047-2_1