Automated Data Labeling for Object Detection via Iterative Instance Segmentation

Jinyoon Kim
School of Science, Engineering, and Technology
Pennsylvania State University Harrisburg
Middletown, PA, USA
juk481@psu.edu

Md Faisal Kabir
School of Science, Engineering, and Technology
Pennsylvania State University Harrisburg
Middletown, PA, USA
mpk5904@psu.edu

***Abstract*—Data labeling in computer vision, specifically in object detection tasks, remains a significant challenge in terms of efficiency and accuracy. This research article introduces an auto-labeling algorithm that combines active deep-learning techniques with the YOLOv8 model. The aim is to automate the data labeling process and enhance the performance of the object detection model. The proposed algorithm automatically labels a portion of unlabeled data based on uncertainty scores, integrating it into the training dataset. This approach reduces the need for manual annotation, which can be time-consuming. The effectiveness of the method is evaluated using two datasets: tomato and apple. The results demonstrate a substantial improvement in the Mean Average Precision score over multiple iterations, highlighting the enhanced performance of the overall model. Moreover, the experiments show that the proposed algorithm surpasses traditional manual annotation methods by generating a higher-performing model with significantly less annotation effort.**

***Index Terms*—Automated Data Labeling, Active Learning, Semi-Supervised Learning, Instance Segmentation, Object Detection**

I. Introduction

Machine learning algorithms can be categorized as either supervised or unsupervised learning, depending on the task of the object [1] and in the case of supervised learning, the availability of extensive, high-quality labeled datasets plays a pivotal role in driving progress especially in the computer vision research. However, obtaining such extensive labeled data is challenging due to high costs and labor-intensive procedures. To mitigate these issues, various approaches have been developed, including Active Learning [2]–[4], Semi-Supervised Learning [5]–[7], and Automated Data Labeling [8]–[10].

Active Learning [11] refers to the use of learning algorithms that proactively choose which data to learn from, aiming to maximize the usefulness gained from each labeled instance. Nevertheless, this method encounters obstacles such as inconsistent performance, sensitivity to hyperparameters, and potential difficulties with certain datasets and tasks.

On the other hand, Semi-Supervised Learning [12] makes use of labeled and unlabeled data, presenting a notable benefit in scenarios where labeled data is limited. It takes advantage of the abundance of unlabeled data, thus reducing the reliance on manual data labeling.

Automated Data Labeling [13] aims to automate the data labeling process either fully or partially, significantly decreasing the time and cost of manual annotation. An example of such a system is the Amazon Automated Data Labeling [10], which simplifies the creation and modification of training datasets. It underscores the drive to streamline the labeling process in complex tasks like instance segmentation [14], significantly impacting the overall effectiveness of computer vision models.

In this study, we integrate diverse strategies, capitalizing on the significant advancements observed in instance segmentation models. A cornerstone of the research is the incorporation of the state-of-the-art YOLOv8 model [15], which is fine-tuned on niche datasets, particularly focusing on plant leaves afflicted with diseases. This tailored adaptation of YOLOv8 magnifies the model's capacity to discern and identify intricate patterns, greatly enhancing the efficacy of auto-labeling algorithms. Further strengthening our approach, we amalgamate this sophisticated technology with the principles of active learning, embedding it into the Automated Data Labeling (ADL) algorithm. The fusion of our iterative training algorithm, combined with the techniques delineated in the preceding paragraphs, yields a marked increase in the performance of the auto-labeling process on an expansive scale. This not only emphasizes the synergy of current technologies but also charts a new trajectory for future research in semi-supervised learning and computer vision.

The notable contributions of this study are as follows:

- Utilized a comprehensive dataset of over 20,000 plant samples for algorithmic training.
- Applied advanced object detection using YOLOv8, enhancing segmentation and accelerating training.
- Implemented a classification correction mechanism alongside an uncertainty data selection algorithm, optimizing model accuracy.
- Introduced semi-supervised learning, leveraging both labeled and unlabeled data for better model performance.
- Incorporated active learning to refine auto-labeling and adjust model weights, ensuring adaptive learning from new data.
- The auto-labeling system serves dual purposes: data annotation and model training, which boosts the accuracy and resilience of object detection tasks.

Following the introduction, Section 2 discusses related

1946-0759/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMLA58977.2023.00124

works on semi-supervised learning, active learning, and the auto-labeling system, offering an understanding of their impact on this research. Section 3 outlines the specific materials and methods employed, including the dataset used for the experiment and a detailed description of the algorithm's step-by-step process. The experimental outcomes and the environmental configuration are presented and analyzed in Section 4. Finally, Section 5 concludes the paper, summarizing the study and suggesting areas for future research improvement.

II. Related Works

This section embarks on a journey through various research studies and approaches that serve as the foundation for the proposed YOLOv8-ADL model. By understanding the valuable insights offered by these works, one can better appreciate the context and rationale behind the design choices in YOLOv8-ADL. The surveyed literature ranges from deep active learning techniques and semi-supervised learning to advanced object detection and weakly-supervised data creation methods.

The comprehensive paper, "A Comparative Survey of Deep Active Learning" [3], shapes the foundation of YOLOv8-ADL with pivotal DAL methods. Pseudo labeling and uncertainty sampling, central techniques in our model, are further enriched by "Deep Active Learning for Named Entity Recognition" [2], despite not adopting its CNN-CNN-LSTM structure. In YOLOv8-ADL, these pseudo labels, produced from predictions, become interim ground truths, while uncertainty sampling zeroes in on complex cases. Seamlessly integrating with YOLOv8's instance segmentation, our model stands at the confluence of deep and active learning, enhancing performance and label accuracy. This unique blend elevates the efficiency of automated labeling, marking a significant leap in machine learning.

The pivotal studies "Semi-supervised Active Learning for Instance Segmentation via Scoring Predictions" [5] and "Learning from Noisy Large-Scale Datasets with Minimal Supervision" [6] have steered advancements in semi-supervised learning. [5] introduced a groundbreaking active learning framework that synergizes initial labeled data with self-labeled data for refined labeling. Meanwhile, [6] emphasized useful insights for the semi-supervised learning, even though its core focus was on noisy annotations that were not concerned in this research. Building upon these foundations, the YOLOv8-ADL algorithm assimilates the robust object detection of YOLOv8 and the spirit of [5]. Unlike [6]'s dual network approach, YOLOv8-ADL champions a singular advanced model. Its active learning loop perpetually hones the model and improves labeled data quality, setting a novel paradigm in processing expansive datasets.

Snorkel, introduced in [9], facilitates rapid training data creation via weak supervision. By allowing users to design labeling functions that generate noisy labels, Snorkel consolidates them into probabilistic labels through a generative model, reducing manual annotation efforts for large datasets. Conversely, our method taps into semi-supervised learning, capitalizing on the YOLOv8 model and the abundance of unlabeled data for training. This direct approach, unlike Snorkel's reliance on human expertise for multiple labeling functions, is more straightforward and demands less human intervention. With the advanced learning capabilities of the YOLOv8 model and minimal labeled data, our method offers precise results, promising a balanced solution in terms of human effort, data utility, and model performance, potentially edging out frameworks like Snorkel in specific contexts.

In overview, the YOLOv8-ADL model reflects the integration and adaptation of several existing methodologies in the fields of active learning, semi-supervised learning, and advanced object detection. It draws from the strengths of these methods, while addressing their limitations to optimize the use of plentiful unlabeled data. The model thus creates a balanced solution, reducing the human effort required in data labeling while ensuring satisfactory performance. Ultimately, the YOLOv8-ADL contributes a new perspective to the ongoing discussions around automated data labeling and machine learning.

III. Materials and Methods

A. YOLOv8 Model Description

The You Only Look Once (YOLO) models [16], [17] revolutionized object detection by unifying location identification and classification, which were traditionally separate steps. This approach allows YOLO models to analyze an entire image during training and prediction in one go. This global view facilitates the recognition of overall contextual patterns within the image, enhancing both the accuracy and efficiency of object detection. The standard architecture of YOLO models comprises three main parts: the backbone for extracting low-level features, the neck for multiscale feature fusion, and the head for final object detection predictions.

Building on this architecture, YOLOv8 [18] introduces notable enhancements. It incorporates an advanced loss function blending Mean Squared Error (MSE) for bounding box regression and Binary Cross Entropy (BCE) for objectness. It also adopts a novel neural network architecture that leverages both Feature Pyramid Network (FPN) and Path Aggregation Network (PAN), significantly boosting prediction accuracy. YOLOv8 offers four model sizes catering to various task requirements and computational resources. In this study, YOLOv8s was employed due to its high performance and computational speed. Moreover, YOLOv8 supports rectangular input shapes and auto-adjusts input size according to the device, enhancing speed for both training and inference. These improvements in YOLOv8 enhance its efficiency, scalability, and adaptability, making it an integral part of the YOLOv8-ADL algorithm for efficient automated data labeling.

B. Dataset Description

The PlantVillage dataset is a publicly accessible collection of leaf images representing various plant species, each labeled with specific disease conditions or as healthy. As detailed in Table I, both PV-Tomato and PV-Apple datasets, derived from

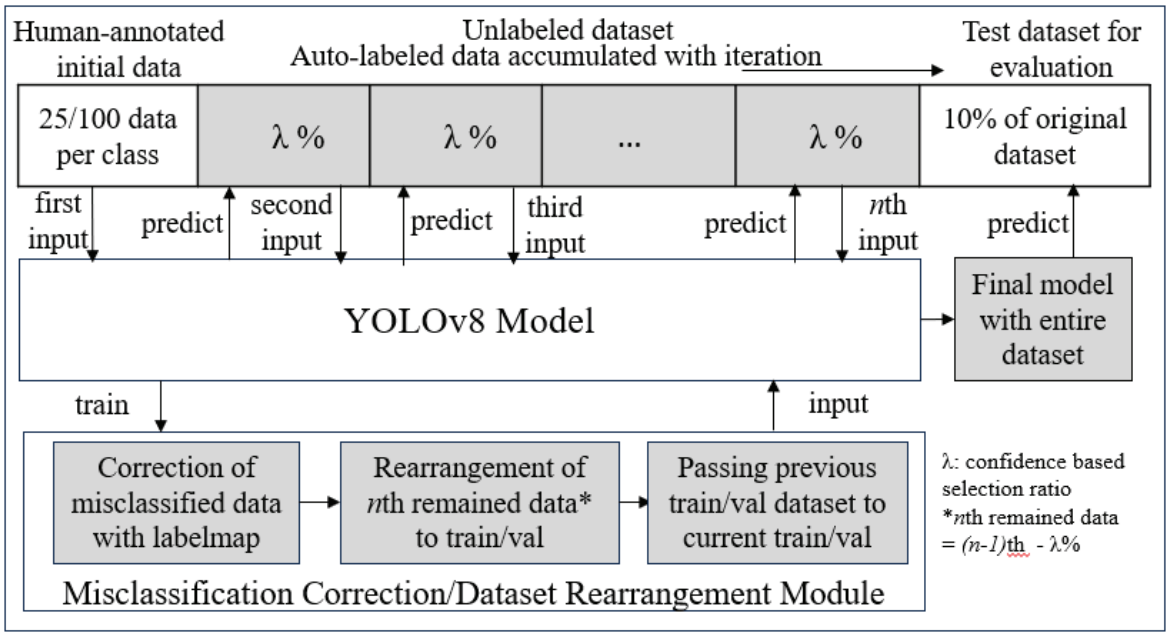

Fig. 1: An overview of the overall architecture.

Dataset	PV-Tomato dataset	PV-Apple dataset
Manual Annotation	Only initial train and test	Entire dataset
Number of images	18160 images	3164 images
Proportion of initial training	25/100 images per class (250/1000 images)	50 images per class (200 images)
Proportion of Active Learning	15348 images	2649 images
Proportion of test data	10% of total (1812 images)	10% of total (315 images)

TABLE I: Plant Vilage Dataset Structure Used for Experiment

Fig. 2: Instance Segmentation annotations delineated by precise polygon shapes, highlighting the distinct texture of the object within the image sample.

the Plant Village [19] dataset, are used in the experiment. The PV-Apple dataset, manually annotated in its entirety, serves as a performance benchmark. In contrast, only the initial training and test subsets of the PV-Tomato dataset are manually annotated, leaving the rest for the algorithm to label. Both datasets have been instance segmented using polygons via the data annotation tool "Roboflow" [20] as represented in Figure 2. This format of annotation, commonly utilized across various studies, was employed not only to bolster the final performance of the model after the training process but also to enable the creation of a comprehensive auto-labeled dataset in the format of instance segmentation. This achievement was made possible due to the inherent instance segmentation capabilities of the YOLOv8 model.

C. Methodology

The YOLOv8-ADL methodology utilizes an iterative active learning process. As each cycle of training, prediction, and labeling unfolds, the model refines its object identification and labeling capabilities. This process progressively enriches the labeled dataset and thereby boosts the model's performance.

1) Active Learning Process and Prediction: The proposed model of automated data labeling, as illustrated in Fig. 1, uses an iterative active learning approach. This process begins by identifying all unlabeled images in the dataset. In each iteration, a percentage (λ) of the unlabeled images, determined by the user, are selected for labeling, effectively establishing a heap for each class in the dataset. Within each iteration, images are processed in batches, with the YOLOv8 model, trained on the available labeled data, predicting labels for these images. Each image is assigned a confidence score, which is an indicator of the model's certainty regarding the assigned

label. These confidence scores play an integral role in the auto-labeling process as they help determine which images are selected for labeling in each iteration.

2) Heap Structure, Auto-labeling, and Label Correction: The algorithm processes images and maintains a heap for each class, a structure that stores essential details such as the file name, class, confidence score, and segmentation masks. The heap size is determined by the λ percent calculated earlier. As predictions are made, any misclassified labels are given top priority within the heap. For each misclassification, the output class is corrected by reverting it to the original class saved prior to the prediction, ensuring the accuracy of the labels. At the same time, predicted labels with the lowest confidence scores, which signify a high level of uncertainty, are added to the heap. These can replace existing entries with higher confidence scores if the heap reaches its maximum capacity. Once the prediction phase is completed, all labels within the heap are used for auto-labeling. This process capitalizes on the segmentation masks generated based on the robustness of the YOLOv8 model, producing detailed instance segmentation with polygons accurately fitting the shape of leaf objects in the image. The freshly corrected and auto-labeled images are then merged into the current labeled dataset for future training iterations, directing the active learning mechanism of the ADL algorithm towards the most complex instances.

3) Model Retraining, Performance Evaluation, and Stopping Conditions: Following the auto-labeling and correction process, the labeled images are removed from the unlabeled data pool, and the model is retrained on the updated labeled dataset. With each iteration, the quantity and diversity of the labeled data used for training increase, thereby allowing the model to continually improve its predictive accuracy. After each retraining cycle, the model's performance is evaluated using a test dataset, providing key performance metrics. These metrics often reveal an improvement in the model's performance over time due to the growing size and diversity of the labeled dataset.

The active learning process is repeated until a stopping condition is met. This could occur when there are no more unlabeled images, or when the model is no longer capable of making further progress in detecting the remaining unlabeled images.The entire process is summarized in the pseudocode (Algorithm 1) that outlines the proposed YOLOv8 ADL algorithm.

Algorithm 1 Concise YOLOv8 ADL Algorithm Pseudo Code

Input: Labeled data D_L, unlabeled data D_U, test data D_T, YOLOv8 model M, ADL parameters, λ
Output: Trained model M, metrics, auto-labels for dataset
LOOP Process :
for each cycle **do**
  Train M on D_L
  Predict labels for D_U with confidence scores
  Define heap size for classes as λ percent of unlabeled images
  for each class **do**
    Maintain a min-heap for predictions
    Replace heap top for predictions with higher confidence
  end for
  Auto-label images from heap using YOLOv8 segmentation
  Correct misclassified labels
  Update D_L and D_U
  Assess M on D_T
end for
return M, metrics, auto-labels

IV. EXPERIMENTS

A. Environmental Set up

1) Experimental Environment: The study employs the NVIDIA GeForce RTX 3070 Laptop GPU and YOLOv8 model, refined with select images from Tomato and Apple datasets. All code is executed in Python via PyCharm, initiating the algorithm's active learning cycle for the auto-labeling process.

2) λ Variable Analysis: The impact of varying λ values (10% and 20%) on the performance of the model is examined using both the Tomato and Apple datasets. The variable λ represents the fraction of the unlabeled dataset that gets auto-labeled and added to the training data at each iteration. By manipulating λ, we observe how different proportions of additional data per iteration influence the model's performance enhancement as the training progresses.

3) Dataset Construction and Analysis: Different training set sizes (25 and 100 images per class) are used for the Tomato dataset to assess how the initial size influences the algorithm. The Apple dataset, initialized with 50 images per class, is used as a performance benchmark against fully manually annotated datasets. Balance between the classes of the training dataset is maintained to avoid model bias and improve generalization capabilities.

4) Performance Evaluation: The original data is partitioned, with 10% forming the test dataset for performance evaluations. The performance metric employed is mAP@0.5 scores, widely acknowledged in the field of object detection for its comprehensive performance appraisal capabilities.

B. Experiment on Semi-Supervising Capability

The adaptability and performance of the YOLOv8 ADL model under different initial conditions and λ values was examined using the PV-Tomato dataset. Initial training conditions included sets of 25 and 100 images per class and λ values were set at 0.1(10%) and 0.2(20%). The mAP@0.5 scores, which are reflective of the model's performance, showed initial results dependent on the size of the training set. The model trained with 100 images per class had a slightly higher initial mAP@0.5 score than the model trained with 25 images per class, due to the larger amount of labeled data available for learning.

Initial data	25 per class		100 per class	
Dataset	λ=0.1	λ=0.2	λ=0.1	λ=0.2
Initial	0.6966	0.6966	0.9165	0.9165
iteration 1	0.9176	0.941	0.9359	0.9486
iteration 2	0.959	0.9557	0.9609	0.9574
iteration 3	0.9395	**0.9631**	0.9494	0.9574
iteration 4	0.9546	0.9613	0.9574	0.9615
iteration 5	0.9457	0.9628	0.9619	**0.9639**
iteration 6	0.9604		0.9598	
iteration 7	0.9622		0.9598	
iteration 8	**0.9637**		0.9598	
iteration 9	0.9625		0.9621	
iteration 10	0.9552		**0.9654**	

TABLE II: mAP@0.5 scores for Tomato(λ =0.1), Tomato(λ =0.2) on each initial data type(25/100)

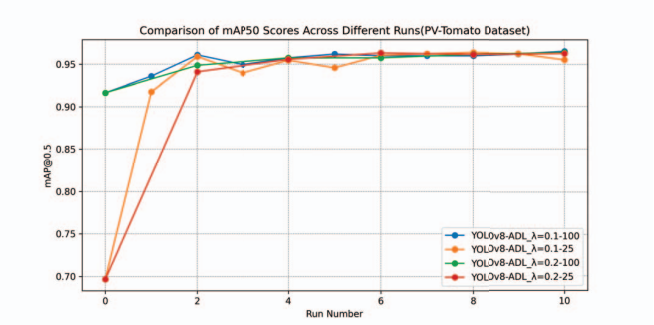

Fig. 3: YOLOv8-ADL performance on PV-Tomato

Dataset	Apple(λ=0.1)	Apple(λ=0.2)	Apple(manual)
initial	0.9083	0.9083	
iteration 1	0.9690	0.9709	
iteration 2	0.9743	**0.9756**	
iteration 3	**0.9751**	0.9625	
iteration 4	0.9728	0.9750	
iteration 5	0.9673	0.9731	0.9649
iteration 6	0.9682		
iteration 7	0.9722		
iteration 8	0.9624		
iteration 9	0.9729		
iteration 10	0.9688		

TABLE III: mAP@0.5 scores for Apple(λ =0.1), Apple(λ =0.2) and Apple(manual) with fixed initial data(50)

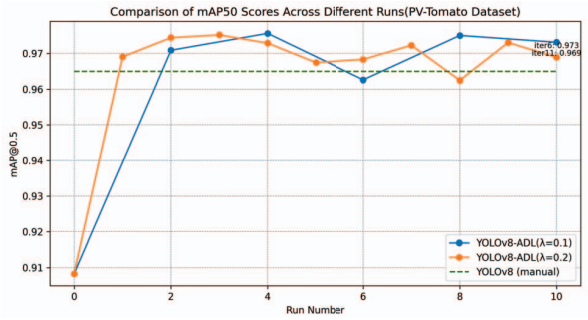

Fig. 4: YOLOv8-ADL and YOLOv8(manual) performance on PV-Apple

Despite this, as the active learning iterations progressed, all models showed steady improvement in their mAP@0.5 scores. Remarkably, the model trained with 25 images per class caught up with the performance of the model trained with 100 images per class, demonstrating the ability of the YOLOv8 ADL algorithm to make effective use of unlabeled data and improve its performance significantly, even when starting with a relatively small amount of labeled data. Specifically, the best performance of the model trained with 25 images per class, as indicated by the highest mAP@0.5 score, was 0.9637 and 0.9631 for λ values of 0.1 and 0.2, respectively. The model trained with 100 images per class achieved a mAP@0.5 score of 0.9654 for λ=0.1 and 0.9639 for λ=0.2. Initially, variations in the λ parameter were anticipated to introduce notable disparities in training progression and accuracy. However, the derived results suggest that changes in λ values did not impart significant differential impacts, emphasizing the model's resilience to such parameter alterations.

These results, encapsulated in Table II and visualized in Figure 3, underline the robustness and adaptability of the proposed algorithm in semi-supervised environments. They underscore the impact of the chosen λ value on the rate of improvement and the number of iterations required for the mAP@0.5 score to converge.

C. Comparison to Human Annotation

The analysis was extended to the Apple dataset, which allowed for an interesting comparison with manually annotated labels. This dataset offered a distinct benchmark to evaluate the performance of the YOLOv8 ADL model. In this scenario, the model began with an initial training set of 50 images per class, using λ values of 0.1 and 0.2, and the mAP@0.5 score was used as a performance metric. The model trained on the fully labeled dataset provided a solid performance with an mAP@0.5 score of 0.9649. This score represents the level of accuracy typically obtained with meticulous human annotation. However, the YOLOv8 ADL model, starting from a lower score, demonstrated a significant ability to improve its performance through active learning. Specifically, the algorithm, when implemented with a λ value of 0.1, progressed to achieve an mAP@0.5 score of 0.9751. An even higher score of 0.9756 was achieved with a λ value of 0.2. These results showcase the capacity of the YOLOv8 ADL model not only to automate the manual annotation process but also to enhance the overall performance of the object detection task.

The variation in the optimal λ value observed across different experiments suggests that this parameter may need to be adjusted based on the specific characteristics of each dataset. This is a valuable insight for future applications of the model, underlining the importance of choosing the right λ value for optimal results. The details of these findings are presented in Table III and visualized in Figure 4. They provide compelling evidence of the model's ability to actively learn from semi-supervised datasets, integrating auto-labeled data into the training set over time, and thereby continually improving its overall performance.

D. Comparison with Other Object Detection Models

The YOLOv8-ADL model, introduced in this study, is evaluated against the models tested in [21] which specifically produced comparable results on tomato dataset from Plant Village in the recent period. Despite potential differences, we strived for comparable environmental settings. In this comparison, YOLOv8-ADL started with an initial set of 100 images per class and a lambda value of 0.1, achieving a mAP score of 0.9654. This surpassed all models in the reference study, including the modified Mask R-CNN. Furthermore, the YOLOv8-ADL model demonstrated efficiencies in time and labor due to reduced human intervention for data annotation. The comparison results are summarized in Table IV. The Apple dataset was excluded because of its lack of study on the subject.

Model Name	mAP on PV-Tomato Dataset
Faster_RCNN	0.625
YOLOv2	0.704
SSD	0.569
YOLOv3	0.731
Mask RCNN	0.882
YOLOv8-ADL(100 images per class, λ=0.1)	**0.9654**

TABLE IV: Comparison of mAP score on PV-Tomato Dataset for each model

These findings suggest that the YOLOv8-ADL algorithm may have a potential advantage over the models discussed in [21] in terms of accuracy, time efficiency, and labor efficiency for the tomato leaf classification task.

V. Conclusion

This study demonstrated the effectiveness of its automated data labeling method in improving the performance of object detection models such as YOLOv8. The results showed that by incorporating a proportion of auto-labeled data into the training set at each iteration, the model's performance, as measured by the mAP@0.5 score, could be significantly enhanced. It can also be noted that the rate of this improvement and the number of iterations required are influenced by the λ value, which dictates the proportion of data to be auto-labeled in each cycle. This implies that the approach could be optimized by carefully selecting the λ value.

The current implementation of the method has proven to be successful in improving model performance. However, there is room for further enhancement. One aspect that could be improved is the mechanism for refining the output labels produced by the model. The addition of a post-processing procedure that refines the label coordinates to be more precise and arranged could potentially boost the model's performance even further. This could involve a process that assesses the shape of the instance segmentation, comparing it against known characteristics of correctly labeled data to identify and remove mislabeled instances.

Additionally, future research could explore the application of this method in other domains beyond object detection, assessing its versatility and effectiveness in different contexts. Furthermore, more advanced techniques could be explored to optimize the selection of the λ value, potentially through adaptive methods that adjust the λ value based on the progress of the model's learning.

To summarize, this method presents a promising opportunity to improve the performance of deep learning models, minimize the manual labor involved in data labeling, and potentially facilitate more precise and efficient object detection. Future work will focus on refining the approach and exploring its broader applicability.

References

[1] M. F. Kabir and S. A. Ludwig, "Enhancing the performance of classification using super learning," *Data-Enabled Discovery and Applications*, vol. 3, pp. 1–13, 2019.

[2] Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod, and A. Anandkumar, "Deep active learning for named entity recognition," *arXiv preprint arXiv:1707.05928*, 2017.

[3] X. Zhan, Q. Wang, K. Huang, H. Xiong, D. Dou, and A. B. Chan, "A comparative survey of deep active learning," 2022.

[4] P. F. Jacobs, G. M. de Buy Wenniger, M. Wiering, and L. Schomaker, "Active learning for reducing labeling effort in text classification tasks," 2021.

[5] J. Wang, S. Wen, K. Chen, J. Yu, X. Zhou, P. Gao, C. Li, and G. Xie, "Semi-supervised active learning for instance segmentation via scoring predictions," 2020.

[6] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 839–847.

[7] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," 2021.

[8] X. Chen, B. Mersch, L. Nunes, R. Marcuzzi, I. Vizzo, J. Behley, and C. Stachniss, "Automatic labeling to generate training data for online lidar-based moving object segmentation," 2022.

[9] A. Ratner, S. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," in *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, vol. 11, no. 3. NIH Public Access, 2017.

[10] "Amazon automate data labeling," 2023, retrieved May 5, 2023. [Online]. Available: https://docs.aws.amazon.com/sagemaker/latest/dg/sms-automated-labeling.html

[11] H. Hino, "Active learning: Problem settings and recent developments," 2020.

[12] Y. Ouali, C. Hudelot, and M. Tami, "An overview of deep semi-supervised learning," 2020.

[13] H. Vishwakarma, H. Lin, F. Sala, and R. K. Vinayak, "Good data from bad models : Foundations of threshold-based auto-labeling," 2022.

[14] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: State of the art," *International Journal of Multimedia Information Retrieval*, 2020.

[15] Roboflow, "Yolov8," retrieved May 11, 2023. [Online]. Available: https://roboflow.com/model/yolov8

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[17] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo: From yolov1 and beyond," 2023.

[18] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with yolov8," 2023.

[19] Kaggle, "Plant village dataset," retrieved May 11, 2023.

[20] R. Inc., "Polygon tool," retrieved May 12, 2023.

[21] P. Kaur, S. Harnal, V. Gautam, M. P. Singh, and S. P. Singh, "An approach for characterization of infected area in tomato leaf disease based on deep learning and object detection technique," *Eng. Appl. Artif. Intell.*, vol. 115, p. 105210, 2022.