Eindhoven University of Technology

Department of Mathematics and Computer Science

In collaboration with VBTI Consultancy B.V.

Semi-Supervised Learning Optimization for Industrial AI Deployment:

A Pseudo-Labeling Framework for Cost-Optimized Computer Vision Pipeline Development

Bachelor Thesis
Data Science

Daniel Osman

Student ID: 1818783

Supervised by,

Illya Kaynov
Dieter Timmers
Maryam Tavakol

July 2025

Contents

Abstract

The success and performance of deep learning models in computer vision tasks heavily depend on the availability of high-quality annotated training data. Despite the importance of annotated data, the process of manually annotating images is both extremely time-intensive and costly, and is traditionally outsourced to external annotators. This process is especially challenging and expensive for complex tasks like instance segmentation with multiple classes and instances. This research investigates the effectiveness of implementing a semi-supervised learning approach using pseudo-labeling techniques to streamline the annotation process while maintaining competitive model performance and reducing costs. This research analyzes the strengths of using a pseudo-labeling pipeline within complex agricultural environments and develops a comprehensive pseudo-labeling structure that iteratively improves model accuracy by using an initial model trained on a small subset of manually annotated data to generate labels on unseen images then retrain. The framework was evaluated on two complex real-world agricultural datasets. The first one uses an asparagus dataset for object detection (12,630 images with 5 classes) and the second uses a cucumber dataset for instance segmentation (2,429 images with 7 classes averaging 70 instances per image). Experiments were conducted separately on each dataset using six different pseudo-labeling strategies tested across five iterations, all with varying balance between ground truth annotations and pseudo-labels. The goal was to determine the effectiveness of pseudo labeling in real world agricultural scenarios and establish optimal supervision thresholds for each use case. The results from the research reveal that pseudo-labeling can achieve substantial cost reductions while maintaining competitive performance. For object detection, the optimal flow (F3) achieved 86.4% of baseline performance while requiring only 43.75% manual annotation effort (56.3% cost reduction). For instance segmentation, flow F4-S achieved 100.5% performance retention with a 75% cost reduction. The segmentation task showed consistently higher efficiency ratios (3.62-43.84) compared to object detection (3.07-5.51), which highlights the value of pseudo-labeling for dense annotation tasks. Manual correction effort decreased by 57.4% across iterations which further illustrates the self-improving nature of the pipeline after each iteration. Beyond the experiments conducted, this research also produced a pseudo-labeling framework integrated within VBTI's OneDL platform. This framework allows employees and clients of VBTI to implement pseudo-labeling across any computer vision project with features like automated database logging, integration with annotation platforms for manual corrections, and support for multiple model architectures. The framework has already been adapted to one of the projects, and several colleagues have shown interest in using the framework for their own real-world applications. VBTI has also shown interest and is considering integrating the framework into their front-end platform. The findings in this paper demonstrate that pseudo-labeling is an effective strategy for reducing annotation costs and accelerating proof-of-concept development in industrial AI applications, specifically for agricultural computer vision tasks.

Chapter 1

Introduction

Over the past few years, there has been a variety of breakthroughs in the field of deep learning and AI which have revolutionized the fields of various industries and have lead to advancements in many developments such as facial recognition, autonomous vehicles, medical imaging, and manufacturing optimization [1]. With access to large datasets and high computational power, deep learning models have been able to achieve high precision and effectiveness in tasks such as object detection, instance segmentation, anomaly detection, and feature extraction. While models can be built for all of these tasks, the success of these models is inherently dependent on the quality and quantity of annotated training data available [1]. Data annotations act as the underlying information embedded within each data entry and serve as the backbone for successful computer vision models [2]. No matter how complex or sophisticated a model's architecture is, without accurate annotated datasets, its performance deteriorates significantly across all metrics and struggles to generalize to unseen data [3].

For object detection and segmentation models in particular, a large amount of high-quality annotations is crucial for achieving robust performance and enabling real-world implementation. Since these computer vision tasks require detailed and precise labeling of objects within an image, such as bounding boxes or instance masks, the annotation process is usually a supervised task, where human annotators manually label data to provide the necessary ground truths for the training models. Despite the development of new annotation techniques, the strong correlation between the accurate annotations and model performance continues to make fully supervised learning the prevailing approach for tasks like object detection and segmentation [4]. Although high-quality annotations enhance a model's capabilities, the annotation process is both time-intensive and costly, and if not done correctly, can introduce inconsistencies that degrade model performance [5].

There have been a variety of research papers and projects involving large image datasets requiring high-quality annotations, all of which encountered the same kind of challenges during the annotation process. The ImageNet project by Russakovsky et al. (2015), intended to develop a high-quality labeled dataset with the goal of enabling the training of more generalizable and accurate computer vision models. ImageNet contained over 14 million images spanning 1,000 object categories, requiring an extreme level of human effort to label and annotate each image accurately. Due to the scale and complexity of the dataset, Russakovsky et al. (2015) had to outsource the labeling process to external annotators, specifically Amazon Mechanical Turk (AMT) [6]. Although this approach accelerated the annotation process and reduced the cost compared to developing an in-house annotation team, it also introduced issues of its own. Firstly, external annotators usually have different levels of expertise and knowledge regarding the data and task being executed. This disparity results in variability in annotation quality which leads to a decrease in model performance and the possibility of requiring extra annotation cycles to ensure high quality [7]. This was apparent in the ImageNet annotation process as the team attempted to mitigate these quality concerns by implementing multiple verification steps, automated validation techniques, and introduced annotation redundancies to ensure that that multiple workers annotated the same images. Although this solution was mostly successful, it increased the complexity of the annotation process, introduced extra costs, and increased the manual labor efforts, which stalled the development process as a whole [6].

A similar research project by Lin et al. (2014) focused on creating the Microsoft COCO dataset for image classification, segmentation, and object detection tasks. While it encountered difficulties similar to the issues presented in ImageNet, the annotation process for COCO was significantly more difficult due to the need for

detailed segmentation masks, which required annotators to precisely trace object boundaries. This additional complexity led to greater inconsistencies across annotations and increased the need for validation cycles to maintain dataset reliability. The Microsoft COCO team also relied on outsourcing, which resulted in annotation disparities and required extensive quality control measures, which further extended the annotation process [8]. It is clear from both research teams that the process of annotating large datasets comes with many inconsistencies but is necessary within the given scope.

In most cases, early-stage development processes do not require as much data as was used by Lin et al. (2014) and Russakovsky et al. (2015). However, while initial development models do not require that much data, startups, established AI organizations, and independent researchers often experience situations where a proof of concept is required under strict deadlines to secure potential clients and investors. Proof of concepts (PoC) are small-scale projects that act as a baseline for AI teams to demonstrate the applicability of their models across a specific business challenge faced by the client [9]. These projects allow clients to assess the viability of an AI solution for their operations and evaluate its potential impact on their business model [9].

As established previously and by Hestness et al. (2017), dataset sizes with high quality annotations tend to consistently lead to improved model performance in image processing through a positive relationship [10]. Therefore, within this industrial setting where early model deployment is crucial, it is essential to annotate as much data as possible to maximize the performance of a model and strengthen the projects potential to attract investors. Since these circumstances revolve around very tight deadlines, developers and annotators are often forced to work long hours to complete labeling tasks as quickly as possible [11]. This increased workload can degrade the annotations produced through annotation fatigue which is a well-documented issue in large-scale manual labeling projects [12]. According to a study by (Parti, 2025), annotators working for extended periods exhibit a higher likelihood of submitting incomplete or inaccurate labels. As a result, instead of improving model performance, rushed or fatigued labeling often degrades the overall quality of the training dataset which directly impacts a model's ability to generalize effectively [12]. A research paper analyzing how errors in annotations affect model performance found that noisy or inconsistent labels lead to unreliable models, increased bias, and weaker generalization across unseen data [13]. Poor labels introduce irrelevant or conflicting information during training, causing the model to learn incorrect features or relationships within the dataset. When implemented, this can lead to mis-classifications, increased false positives, and reduced confidence in the system's predictions, which is especially problematic in domains like autonomous driving, medical imaging, and industrial automation.

To address these challenges, this research investigates the effectiveness of implementing a semi-supervised learning approach to streamline the annotation process for object detection and segmentation tasks by using pseudo-labeling techniques. By utilizing these pseudo-labeling techniques, it is possible to create a cycle that continuously improves annotations by using an initial model trained on a small subset of data to generate labels on unseen images, refining those labels manually, then retraining to ultimately end up with a model that becomes increasingly accurate and efficient over time. The goal is to evaluate whether this pseudo-labeling pipeline can reduce annotation time while maintaining quality, improve model accuracy, and enhance the proof-of-concept development process within the field of AI or the VBTI environment.

Chapter 2

Related Work

2.1 Pseudo-Labeling in Computer Vision

Numerous AI development projects require large volumes of consistently labeled data to achieve peak model performance. Although the concept of pseudo-labeling has always existed in machine learning, Lee (2013) formally introduced it as a technique to improve model performance through retraining, establishing a foundation for transforming the annotation process from fully supervised to semi-supervised. The process included the idea of increasing a model's performance through adding unseen data to the training set, which was labeled using an initial model trained on data with ground truth annotations. This method proved to be efficient as a way of allowing the model to iteratively refine itself over a number of retraining processes. Following this established research by Lee (2013), pseudo-labeling became widely studied across the field of AI, with researchers consistently evaluating its effectiveness when implemented in practice [14]. The approach consistently involves iteratively assigning pseudo-labels to unlabeled data with confidence scores above a set threshold, expanding the labeled dataset, and then retraining the classifier. This process has proven to be particularly valuable in applications where manual annotation is either very time consuming or very expensive, and has been expanded to serve as a method for bootstrapping and initializing AI-based projects.

2.1.1 Object Detection

To improve the annotation process for object detection datasets, Hu et al. (2022) applied pseudo-labeling within a framework designed to detect motion cues in road videos in an attempt to improve autonomous driving. Their study used a large-scale driving dataset containing annotated video sequences of urban and highway environments. The researchers first trained an initial object detection model on a small manually labeled subset of the full dataset. The trained model then generated pseudo-labels for the remaining unlabeled frames, and this process was consistently repeated until the model performance stabilized. The study found that the addition of pseudo-labeling into the training pipeline led to significant improvements in object detection accuracy, with their PseudoProp method outperforming other semi-supervised object detection methods by 7.4% on mAP75 when tested on the large-scale Cityscapes dataset. This improvement was achieved by expanding the training dataset using higher-quality pseudo-labels and using fusion techniques to reduce noise and improve the reliability of the generated pseudo-labels. Considering the time and cost requirements that arise with annotating such detailed video frames, this suggests that pseudo-labeling can be a valuable tool across different types of data if applied in a streamlined manner [15].

2.1.2 Semantic Segmentation

Pseudo-labeling has also been extensively explored in semantic segmentation which requires precise pixel annotations rather than simple bounding boxes. Due to the nature of segmentation labels, the annotation process is even more time-consuming. PseudoSeg was a framework introduced by Zou et al. (2020) specifically for semi-supervised semantic segmentation. It incorporated a structured label refinement process that ensured pseudo-labels maintained high consistency throughout training. Using this framework, with the addition to a confidence-based filtering mechanism, the model was able to iteratively improve its own predictions while minimizing errors introduced by incorrect labels. One of the experiments conducted revolved around applying

the framework to a widely bench-marked dataset for object detection and semantic segmentation (PASCAL VOC 2012) [16]. When applied, Zou et al. (2020) found that the pseudo-labeling model outperformed the baseline model by 9.13 in mean Intersection over Union. The baseline model was trained on only 1/8 of the labeled data, and the metric used measures the average overlap between the predicted segmentation and the ground truth across all classes. Given the complexity of segmentation tasks, these findings suggest that pseudo-labeling can potentially improve model performance, making it a viable strategy for large-scale datasets [17].

2.1.3 Agricultural Computer Vision Applications

Pseudo-labeling has also shown particular promise in agricultural computer vision applications, where annotation processes require domain expertise and consideration of precision-recall tradeoffs. Ferreira et al. (2023) evaluated pseudo-labeling techniques for animal identification by training deep neural networks to identify Holstein cows within agricultural environments. Despite their dataset being relatively clean with clear visual patterns and minimal environmental noise, when they applied pseudo-labeling to their dataset, the researchers observed a significant increase in accuracy of up to 20.4% compared to training models using only limited manually labeled images. Their approach demonstrated that pseudo-labeling could effectively leverage unlabeled agricultural data to expand training sets while simultaneously improving overall model performance. The final model achieved 92.7% accuracy when tested on an independent test set containing 59 individual cows [18].

2.2 Pseudo-Labeling Beyond Computer Vision

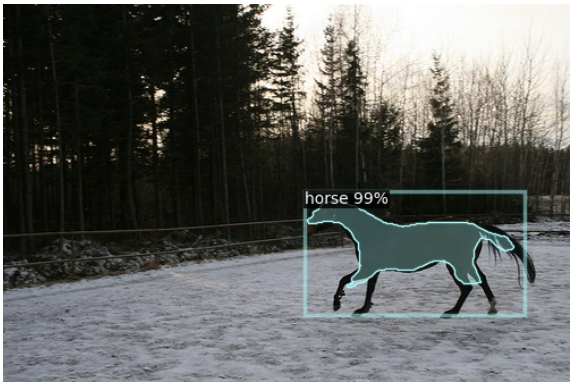
While this research focuses on computer vision applications within real world environments, pseudo-labeling has evidently been adapted across diverse domains that go beyond the field of computer vision. This process has been proven effective across numerous research papers as a cost-effective solution for reducing manual annotation requirements.

2.2.1 Pseudo-Labeling in Natural Language Processing

Recent papers in natural language processing have showcased how pseudo labeling has the potential for training large language models without extensive human annotation. This was on display when Wang et al. (2022) introduced Self-Instruct which was a framework created with the intention of improving instruction-following capabilities of pretrained language models by bootstrapping off their past models [19]. Their approach generates instructions, input, and output samples from a language model, filters invalid samples, then uses correct ones to fine-tune the original model. This framework was applied to the vanilla version of GPT-3 where it demonstrated a 33% absolute improvement over the original model on Super-Natural Instructions, achieving performance on the same level as InstructGPT-001, which was trained with private user data and human annotations. This framework represents a relevant example for this research because it demonstrates how pseudo-labeling can achieve near-human-level performance while dramatically reducing annotation costs. The method provides an almost annotation-free approach for aligning pre-trained language models with instruction processes, which almost directly reflects some of the goals of this thesis in the computer vision domain. Since the introduction of Self-Instruct, instruction-following models trained through pseudo-labeling techniques have become a slowly emerging aspect of modern language model development, with major AI systems (like ChatGPT and Claude) now relying on similar self-training methodologies.

2.3 Limitations and Challenges in Current Approaches

While the research by Zou et al. (2020), Hu et al. (2022), and Ferreira et al. (2023) all established solid foundations for pseudo-labeling effectiveness, several limitations emerge when considering real-world deployment scenarios [15, 17, 18]. Most existing pseudo-labeling studies have been conducted on controlled benchmark datasets like PASCAL VOC or standardized driving datasets, where data variability is limited and environmental conditions are relatively predictable. The issue arises when dealing with images with poor visibility, unstable resolution, and within real world industrial scenarios.



(a) PASCAL VOC [16]

(b) YL Strawberry Dataset

(c) PASCAL VOC

(d) VDL Cucumber Dataset

Figure 2.1: Comparison of dataset complexity: PASCAL VOC benchmark dataset (left) versus real-world agricultural data from VBTI (right).

Figure 2.1 sets a good base for the drastic difference in complexity between benchmark datasets like PASCAL VOC and real-world agricultural data. The images on the right are clearly much more complex since they are captured from real greenhouse environments with varying lighting conditions, overlapping vegetation, and contain over 70 instances per image compared to the relatively simple and clean scenes found in benchmark datasets. The effectiveness of pseudo-labeling in uncontrolled, real-world environments with high data variability is still largely unevaluated. Task-specific challenges also vary significantly across different computer vision applications. While pseudo-labeling may work effectively for object detection in structured environments, its performance on dense and overlapping instances in agricultural settings has not been thoroughly investigated. Additionally, the scalability of pseudo-labeling approaches for industrial deployment presents practical challenges that academic studies have not yet addressed. Most current pseudo-labeling frameworks lack the streamlined integration that is necessary for real-world annotation pipelines, which creates significant barriers for companies looking to adopt these techniques. Furthermore, existing research has primarily focused on performance improvements without thoroughly analyzing the cost savings potential that pseudo-labeling could provide for industrial annotation workflows. Without addressing these implementation challenges and economic considerations, the potential benefits of pseudo-labeling remain difficult to test within agricultural and industrial companies (like VBTI).

Chapter 3

Research Question & Primary Focus

3.1 Focus

To address the limitations discussed in the previous section, this study focuses on evaluating pseudo-labeling effectiveness for performance and cost optimization in niche agricultural datasets within the VBTI environment. Although there exists commercial platforms like Encord, Supervisely, and SuperAnnotate who offer pseudo-labeling capabilities, there still remains limited academic research on the specific cost-performance trade-offs and optimal supervision thresholds for complex agricultural datasets. Unlike prior studies, which often relied on benchmark datasets or generalized scenarios, this research uses hyper-specific datasets provided by VBTI, which are directly derived from industrial use cases. These datasets reflect the complex challenges encountered in real-world AI applications, such as agricultural automation and manufacturing environments, making them a more challenging baseline for evaluating the scalability and reliability of pseudo-labeling strategies. This real-world aspect not only sets this work apart from existing literature but also ensures that it is practical and relevant to industrial scenarios while also being production ready.

Furthermore, while human loop approaches exist in pseudo-labeling literature, most studies focus primarily on performance improvements rather than systematically analyzing the economic trade-offs between manual correction effort and model performance. Although manual correction increases the level of supervision, it has the potential to produce a higher-performing initial model and lead to more consistent annotations over time. This research builds upon the pseudo-labeling strategies proposed by Zou et al. (2020) and Hu et al. (2022), and extends them by systematically evaluating how different supervision strategies impact both annotation costs and model performance in agricultural datasets, ultimately aiming to reduce the need for outsourcing the annotation process to external parties. The goal is to determine optimal supervision thresholds that maximize cost savings while maintaining competitive performance, providing a pathway for organizations to accelerate proof-of-concept development while controlling annotation budgets.

The goal is to refine the annotation process to improve efficiency, accuracy, and scalability while reducing manual effort. When optimized, the framework will aim to reduce annotation time, improve label quality, and enhance proof-of-concept development across the AI industry. Given the established goals and aims of the research, it is crucial to evaluate whether this approach can effectively balance automation with annotation reliability while maintaining high model performance. Therefore, the following research points have been developed:

3.2 Research Question

"How effective is a streamlined pseudo-labeling pipeline in improving annotation efficiency and model performance across niche datasets, and how does it impact proof-of-concept development in terms of manual annotation costs and turnaround time?"

3.2.1 Sub-Research Questions

Sub-Question 1: Comparing Fully-Supervised and Pseudo-Labeling Frameworks for Model Effectiveness

"How does a pseudo-labeling workflow compare to a fully supervised annotation workflow in terms of model performance, training efficiency, and deployment viability?"

This sub-question aims to evaluate the final performance of models trained entirely on ground truth vs. those trained entirely on pseudo-labels. The research will compare the performance metrics, model convergence, and quality between both pipelines while analyzing the trade-offs between costs, workloads, and performance.

Sub-Question 2: Determining the Ground Truth Threshold for Effective Pseudo-Labeling

"How does training the initial model with varying amounts of fully annotated ground truth data affect downstream model performance and pseudo-label stability?"

This sub-question will measure model performance (mAP, F1, precision, recall) across pipelines initialized with different amounts of ground truth data (e.g., 50, 150, 250, 500 samples). The research will assess how the early number of ground truth annotations influences the quality and consistency of generated pseudo-labels in later iterations and explore task-specific differences between object detection and segmentation in terms of minimum viable ground truth requirements.

Sub-Question 3: Evaluating the Practical and Economic Efficiency of Pseudo-Labeling Workflows

"How practical is the pseudo-labeling pipeline in real-world annotation workflows, and how does it compare to manual labeling in terms of correction effort, cost, and speed?"

This sub-question will look into the economic trade-off between ground truth annotation effort and model quality and will compare scenarios with different ground truth-vs-pseudo labeled balances. The goal is to find the most optimal balance between ground truths and pseudo labels with competitive performance and sufficient cost optimization. Additionally, the analysis will examine how difficult pseudo-labels are to correct by analyzing annotation time, correction density, and effort across iterations to determine the practicality of the framework.

Chapter 4

Methods and Approach

4.1 Pseudo-Labeling Pipeline

The goal of this research is to evaluate the performance of pseudo-labeling techniques in niche real-world datasets and assess their cost-effectiveness for reducing annotation expenses in agricultural applications while maintaining competitive model performance. To accomplish this evaluation, a streamlined pseudo-labeling pipeline was developed as an experimental tool to systematically test different supervision strategies across object detection and segmentation tasks. The pipeline must be designed to iteratively improve model development until it reaches a threshold capable of outputting annotated data at a high rate of accuracy. It should be capable of generalizing across both instance segmentation and object detection tasks, and should be reusable and adaptable to a variety of different datasets and annotation guidelines.

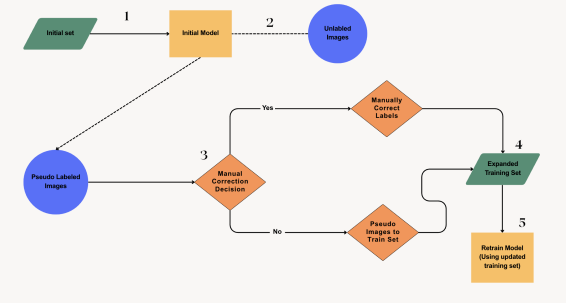


Figure 4.1: Pseudo-Labeling Pipeline

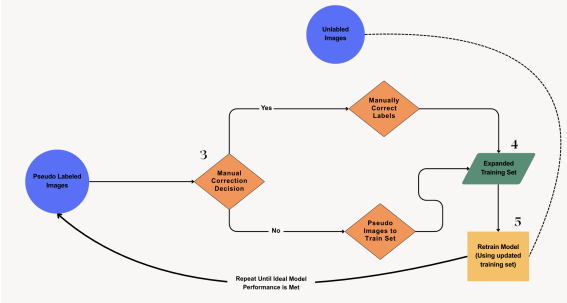


Figure 4.2: Pseudo-Label Retraining Loop

To support the implementation of this project and evaluate the effectiveness of an overall streamlined pseudo-labeling process, Figures 4.1 and 4.2 show a visual representation of how the pipeline will be designed to function. This diagram outlines the full structure of the iterative workflow and consists of six main sections that are all interconnected and each represents a distinct stage in the annotation and model training process. These six stages work together to enable the pipeline to gradually improve model performance and therefore annotation quality.

4.1.1 Steps

The six primary sections of the pipeline are as follows:

1. **Initial Model Setup & Training:** In order to first initiate the pseudo-labeling cycle, a small subset of data (≈50) images must first be manually annotated and used to establish the initial training set for the first iteration of the detection model. Once established, the initial model will be trained on this small training set and will be later used in the upcoming steps. The model architecture will be pre-defined and

will depend on the task being executed. For this research, the models used will either be Faster R-CNN or Mask R-CNN depending on whether the task involves object detection or instance segmentation.

2. **Pseudo-Label Generation:** After the initial model is trained and the training set has been established, a separate subset of unannotated data is sampled from the database, and the initial model is used to run inference on that data to generate pseudo-labels for the instances. This step results in an additional set of images, each accompanied by corresponding pseudo-labels.

3. **Manual Correction Phase:** After running inference on the new subset of unlabeled data, the images and their pseudo-labels are reviewed and corrected within an annotations platform to eliminate inaccurate labels. This step ensures that incorrect or missed instances are corrected and that noisy labels are removed before the next step of the pipeline. Throughout the manual correction phase, it is also possible to utilize AI assisted annotation methods (SAM) to further expand on the basis of automation.

4. **Dataset Expansion:** The images and their corrected labels are then exported from from the annotations platform and merged with the existing training set and prepared for the next iteration of model training.

5. **Model Training Iteration (x):** The updated and expanded training set, which now includes both the original manually labeled images and the newly corrected pseudo-labeled data, is then used to train the next iteration of the model. This process is treated as a full retraining step which means the model is trained from scratch using the expanded training set. This approach helps with removing errors from earlier predictions and allows the model to better generalize across unseen images. After many cycles and successive iterations of this process, this retraining phase will gradually improve model performance and annotation accuracy.

6. **Evaluation (x):** After each training iteration, the model is evaluated on an external test set that is not used during training or validation. This step is meant to assess the model's ability to generalize to unseen data and measure its overall performance. If the model fails to meet a certain performance threshold based on precision, recall, or mAP, the pipeline cycle is repeated and an additional batch of unlabeled data is introduced to further expand the training set. This essentially means repeating steps 2-5.

7. **Automated Training Expansion:** In instances where the pseudo-labels demonstrate sufficient accuracy and consistency, it becomes possible to transition the pipeline to become fully automated. When the model reaches acceptable performance thresholds, the pipeline can skip the manual correction step and continue expanding the training set using only pseudo-labels from previous iterations. This automated expansion continues until model performance stabilizes or reaches the desired accuracy for deployment. Unlike the approach used by Hu et al. (2022) [15] and Zou et al. (2020) [17], who relied entirely on automated pseudo-labeling throughout their processes, this framework incorporates initial manual correction phases to ensure higher label quality before transitioning to full automation.

4.2 Mathematical Framework

The pseudo-labeling pipeline can be formalized as an iterative optimization process. Let D_0 be the ground truth dataset with approximately x samples. At each iteration i, the ground truth data is combined with the pseudo-labeled data to train a new model:

$$\theta_i = \arg\min_{\theta} \mathcal{L}(D_0 \cup D_{pseudo}^{(i)}, \theta) \tag{4.1}$$

where:

$$\begin{aligned} \theta_i &= \text{model parameters at iteration } i \\ D_0 &= \text{ground truth dataset} \\ D_{pseudo}^{(i)} &= \text{pseudo-labeled data accumulated up to iteration } i \\ \mathcal{L} &= \text{loss function used to train the model} \end{aligned}$$

Iterative Dataset Refinement

After training an improved model θ_i, that model is then used to generate new pseudo-labels for all unlabeled data, replacing any previous pseudo-labels to ensure consistency:

$$D_{train}^{(i)} = D_0 \cup f_{\theta_i}(D_{unlabeled}) \cup D_{corrected}^{(i)} \tag{4.2}$$

where:

$$\begin{aligned} D_0 &= \text{initial ground truth dataset} \\ f_{\theta_i}(D_{unlabeled}) &= \text{pseudo-labels generated by model } \theta_i \text{ on unlabeled data} \\ D_{corrected}^{(i)} &= \text{manually corrected labels from all previous iterations} \end{aligned}$$

Cost Analysis

The total annotation cost for the pseudo-labeling approach is:

$$\boxed{\text{Total Cost} = C_{initial} + \sum_{i=1}^{T} N_{corrections}^{(i)} \cdot C_{per\_correction}} \tag{4.3}$$

where:

$$\begin{aligned} C_{initial} &= \text{cost of initial manual annotations} \\ N_{corrections}^{(i)} &= \text{number of corrections needed at iteration } i \\ C_{per\_correction} &= \text{cost per manual correction} \\ T &= \text{total number of iterations until convergence} \end{aligned}$$

Note that generating pseudo-labels is only computational which is why this process is highly cost-effective compared to full manual annotations.

4.3 Experimentation Strategy

In order to evaluate the effectiveness of the pseudo-labeling framework outlined in the previous section (4.1) and determine the threshold for a successful implementation of this process, two separate experiments will be conducted. The first experiment consists of six controlled flows that gradually reduce the amount of manual correction across each iteration. The second experiment focuses on manual correction effort and is used to estimate annotation cost based on correction counts. Both experiments are designed to test the framework under different levels of supervision and provide insights into performance, efficiency, and practical viability. For the sake of simplicity, manually corrected data will be referred to as ground truth data.

4.3.1 Multi-Flow Experiment Design

The first experiment consists of testing six different pseudo-labeling flows (F0–F5), each of which represent a different balance between images with ground truth annotations and images with pseudo annotations. The purpose of this experiment is to estimate how much ground truth data is needed in the early stages in order to achieve reliable pseudo-labeling performance in later iterations. Each flow will run for 5 iterations and will start from the same initial model, trained on 50 manually annotated ground truth (GT) images. After this, each flow will add 150 new images per iteration, but will vary in whether those new images are manually corrected (GT) or model generated pseudo-labels (PL).

The breakdown for each flow is summarized in the following table:

Table 4.1: Flow breakdown across 5 iterations.

Flow ID	Iteration Breakdown	Description
F0	[GT, GT, GT, GT, GT]	Full ground truth, serves as control line
F1	[GT, GT, GT, GT, PL]	4 iterations manual, final one is pseudo-labeled
F2	[GT, GT, GT, PL, PL]	3 manual, 2 pseudo
F3	[GT, GT, PL, PL, PL]	2 manual, 3 pseudo
F4	[GT, PL, PL, PL, PL]	1 manual correction, rest pseudo
F5	[PL, PL, PL, PL, PL]	All pseudo-labels after initial model

At each iteration, the training set is expanded by 150 new images, which are either ground truth (GT) or pseudo-labeled (PL) depending on the flow/iteration. After every expansion, the newly updated dataset is used to retrain the model to evaluate how performance changes over each iteration/expansion of the training set. The total number of training images per flow after each iteration is shown in the table below:

Iteration	Cumulative Training Images
0	50
1	200
2	350
3	500
4	650
5	800

Table 4.2: Image count per iteration (50 + 5×150).

Each flow is evaluated at every iteration by training a new model on the cumulative dataset and testing it on an external evaluation set. Model performance is then measured using a combination of metrics, including mAP@50, mAP@75, and overall mAP (mean average precision). In addition to mAP, standard classification metrics such as F1 score, precision, recall, false positives (FP), false negatives (FN), and true positives (TP) are also tracked to provide a more complete view of model behavior across different supervision levels. These metrics allow us to compare how different flows perform as the training data grows over time.

4.3.2 Economic Annotation and Manual Correction Analysis

This experiment focuses on analyzing both the cost reductions in the multi-flow experiments and the efficiency of a manual correction phase. Instead of focusing strictly on model performance, the goal here is to simulate a realistic correction cycle, where pseudo-labeled predictions are manually reviewed and adjusted over multiple iterations. At each step, the number of corrections per image is recorded, along with the time spent per correction. During this experiment, the following are logged per iteration:

- Number of corrections made (per bounding box)
- Time spent correcting pseudo-labels
- Corrections per image
- Final training time for each model

This data can be used to estimate the annotation costs related to manual corrections by multiplying correction counts by the price per annotated instance. By conducting the 6 controlled flows along with investigating annotation costs associated with the whole process, it becomes possible to observe how different levels of manual supervision impact model performance, annotation effort, and overall annotation costs. These two experiments aim to display a deeper level of analysis on the effectiveness of pseudo-labeling on model performance, and helps to address all the sub-questions in context.

4.4 Use-Cases

The experiments will be conducted using two different use cases. The first use case involves an object detection dataset, while the second involves a instance segmentation task. Both datasets are niche and were collected for agricultural applications, which means that accurate detection is critical for satisfactory results. By using these datasets as two separate use cases for the experiments, it becomes possible to assess the overall effectiveness of pseudo-labeling on complex datasets, while also providing VBTI with insights into how pseudo-labels can be utilized to achieve high performance to reduce overall labeling costs.

- **Use Case 1 – Object Detection:** The first use case is an object detection task using the AVL Asparagus dataset. This dataset consists of approximately 80,000 high-resolution field images that capture asparagus vegetables emerging from the soil. Each image contains an average of three bounding box annotations and three ellipse annotations, with five classes: *defects*, *above ground*, *tips*, *overgrown*, and *stones*. Based on current annotation service rates that VBTI has with [redacted], the estimated cost for manually labeling the entire dataset would be approximately €15,000. In addition to these high costs, the annotation provider is only able to deliver 800 to 1,000 annotated images every two weeks. This slow turnaround essentially means that annotating the entire dataset through the annotation provider could easily take a number of weeks. Due to these constraints, this dataset acts as an ideal candidate for the current research.
- **Use Case 2 – Instance Segmentation:** The second use case involves the VDL Cucumber dataset which consists of segmentation masks. The dataset includes 2,429 high-quality images with seven segmentation classes: *Cucumber*, *Main Stem*, *Cucumber Stem*, *Leaf Stem*, *Node*, *Plastic Ring*, and *Metal Ring*. On average, each image contains around 70 annotated instances, with individual annotation prices ranging from €0.0555 to €0.1725 depending on the class. The estimated cost of manually labeling the entire dataset would be approximately €13,000, and since segmentation masks are complex and difficult to annotate, the cucumber dataset is an ideal candidate for the experimentation process, as its complex and has clear cost-saving potential.

Both use cases serve to test the adaptability of the pipeline to different annotation tasks, and help assess the overall framework when applied to real world datasets. The findings from these implementations will directly inform the design and development of the final pipeline interface.

Chapter 5

Use Case 1: Object Detection

5.1 Overview - Asparagus Dataset

The first use case revolves around the AVL Motion project, which was developed by VBTI to provide a vision based solution for detecting and localizing asparagus in open environments. The primary goal was to enable a robotic harvesting machine to accurately detect various growth stages of asparagus with minimal error to accurately harvest at maximized rates. The main requirement for such a machine is a custom built vision system capable of identifying the asparagus crowns within soil-heavy images, estimating their precise base location, and minimizing false detections to avoid damaging the field or missing viable crops. The resulting system utilized a two-stage deep learning pipeline for object detection, but like many vision applications, it heavily depended on accurate annotations which are extremely critical and time-intensive. Despite the fact that the full dataset collected by VBTI consists of approximately 80,000 images of asparagus emerging from the soil, this work utilizes a representative subset of 12,630 annotated instances for the experiments.

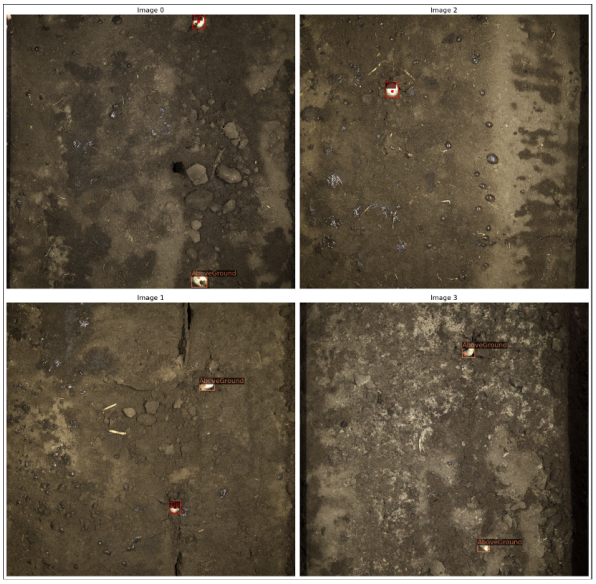


Figure 5.1: Image from the AVL Asparagus dataset

The dataset comprises five distinct classes: *defects*, *above ground*, *tips*, *overgrown*, and *stones*. There is a clear class imbalance, with *Defect* being the most common at 12,314 instances, followed by *AboveGround* with 10,521, and *Tip* with 3,805 instances. The least frequent classes are *Overgrown* and *Stone*, with 2,246 and 751 instances, respectively (A.1). On average, each image contains around 1 to 3 instances, as shown by the distribution of instances per image. The images are relatively clear despite being collected in natural agricultural environments. They were captured during the 2023 asparagus harvesting season using a stereo-camera system

and stored at a resolution of 3200×3000 pixels and all depict the field from a top-down view which shows one or more asparagus spears emerging from the soil. Given the requirement for precise annotations and, the cost and time constraints associated with manual annotation process, this use case is ideal for evaluating the potential time and cost savings involved with utilizing the proposed pseudo-labeling pipeline.

5.2 Test Set

For the purpose of this research, a test set was created comprising of 150 images to ensure that all evaluations of model performance were reliable across all experiments. The purpose of this set is to provide an unbiased evaluation metric for comparing different models and allow for consistent performance tracking as models evolve through successive training iterations. The set was selected to be representative of the overall dataset while remaining completely separate from all training data used in the experiments. This separation is necessary for maintaining performance as it ensures that the test data does not influence the pseudo labeling process or the process of model training in general. The 150 images contain a total of 210 annotated instances distributed across all five classes: *above ground*, *tip*, *defect*, *overgrown*, and *stone*. Using this test set, it is possible to extract consistent mAP@50, mAP@75, precision, recall, and F1-score metrics across all experimental flows, which is essential for comparing the effectiveness of pseudo-labeling within our framework.

5.3 Chosen Model Architecture

The model architecture that will be used for the object detection pseudo-labeling process will be a **Faster R-CNN** with a **RegNet-GF3** backbone. Faster R-CNN is a two-stage object detection framework that first proposes candidate object regions using a Region Proposal Network (RPN), and then classifies and refines these proposals using a detection head [20].

The general output of a Faster R-CNN model is based on the following formulation:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{5.1}$$

where:

- L_{cls} is the classification loss (typically cross-entropy),
- L_{reg} is the regression loss (usually smooth L1),
- p_i is the predicted probability of anchor i being an object,
- p_i^* is the ground truth label (1 for object, 0 for background),
- t_i and t_i^* are the predicted and ground truth bounding box coordinates,
- λ is a balancing parameter.

The RegNet-GF3 backbone was selected because of its efficient design and strong feature extraction capabilities, especially when dealing with medium-to-high-resolution imagery. Based on the RegNet introduced by Radosavovic et al. [21], RegNet-GF3 achieves a balanced trade-off between complexity and being representational which makes it suitable for 3200×3000 pixel images in agricultural environments. Dealing with high-resolution images comes with many challenges such as detecting small-scale objects and identifying ground truths within images that have visually complex backgrounds. In datasets like this, many images contain only a few labeled instances despite having a large pixel area. It has been discussed in various papers that large, high-resolution images containing only a few annotated objects often lead to missed detections and an increased number of false positives [22]. For such reasons, the model architecture chosen must be capable of adapting to the spatial sparsity and variability present in these types of images [23], making the Faster R-CNN with a RegNet-GF3 backbone an ideal choice in this case. In the context of applying the pseudo-labeling pipeline to the asparagus dataset, all iterations of the framework will be trained using this architecture, with consistent hyperparameters to ensure fair comparison across each flow and iteration:

- **Architecture**: Faster R-CNN with RegNet-GF3-2 backbone

- **Training epochs**: 50 per iteration
- **Batch size**: 6
- **Optimizer**: Default SGD optimizer
- **Data augmentation**: None (to maintain annotation consistency)
- **Retraining approach**: Complete model retraining from scratch each iteration

5.4 Experimentation Results

5.4.1 Initial Model Setup and Baseline Performance

All flows began with an identical initial model which was trained on 50 manually annotated images with the model architecture discussed in section 5.3. This initial model achieved a baseline mAP@50 of 0.177, recall of 0.771, and precision of 0.346, which shows its relatively below average performance given the limited training data and standard model hyperparameters. For reference, all model results can be seen in table A.1 within the appendix. While the initial models' performance is suboptimal and can be improved, it serves as a baseline that demonstrates the potential for improvement through using pseudo-labeling techniques. Assuming that some training configurations are enhanced (increased epochs, optimized learning rates, or advanced loss functions), these baseline results could be substantially improved. This assumption essentially leads to the establishment that the findings presented represent a lower bound on the effectiveness of pseudo-labeling for this specific dataset.

5.4.2 Multi-Flow Performance Analysis

The multi-flow experiment, as explained in chapter 4, looks into six different pseudo-labeling strategies (F0–F5) across five iterations, with each flow having different levels of ground truth annotations. Table 5.1 displays the experimental design in detail, where GT represents ground truths (or manually corrected) annotations and PL represents pseudo-labels generated by the model. Note that all experimental flows and iterative training procedures described in this section have been conducted in accordance with the mathematical framework outlined in Section 4.2.

Table 5.1: Flow breakdown for Object Detection

Flow ID	Iteration Breakdown	Description	Model Type
F0	[GT, GT, GT, GT, GT]	Full ground truth, control line	Faster R-CNN (Object Detection)
F1	[GT, GT, GT, GT, PL]	4 iterations manual, final one pseudo-labeled	Faster R-CNN (Object Detection)
F2	[GT, GT, GT, PL, PL]	3 manual, 2 pseudo	Faster R-CNN (Object Detection)
F3	[GT, GT, PL, PL, PL]	2 manual, 3 pseudo	Faster R-CNN (Object Detection)
F4	[GT, PL, PL, PL, PL]	1 manual correction, rest pseudo	Faster R-CNN (Object Detection)
F5	[PL, PL, PL, PL, PL]	All pseudo-labels after initial model	Faster R-CNN (Object Detection)

After conducting all experimental flows, Table 5.2 presents the performance metrics for all flows at their final iteration (Iteration 5). This table provides a direct comparison of how different supervision strategies impact final model performance.

Table 5.2: Final Performance Comparison Across All Flows (Iteration 5)

Flow	GT	PL	mAP@50	mAP@75	mAP(All)	Prec.	Recall	F1
F0	800	0	0.484	0.462	0.374	0.402	0.842	0.544
F1	650	150	0.428	0.410	0.333	0.432	0.849	0.573
F2	500	300	0.372	0.324	0.276	0.485	0.851	0.618
F3	350	450	0.418	0.370	0.309	0.473	0.832	0.603
F4	200	600	0.308	0.289	0.223	0.527	0.829	0.644
F5	50	750	0.144	0.042	0.073	0.525	0.689	0.596

Performance Interpretation

As stated previously, all flows began with identical initial models trained on 50 manually annotated images with mAP@50 of 0.177. This model had relatively low performance which was expected given its limited training data. The results from Table 5.2 reveal a clear relationship between the amount of ground truth data and model performance. Flows with higher amounts of ground truth data (F0, F1, F2) had more consistent performance improvements, while flows with limited manual supervision (F4, F5) were not as good.

The relationship can be seen properly in Figure 5.2:

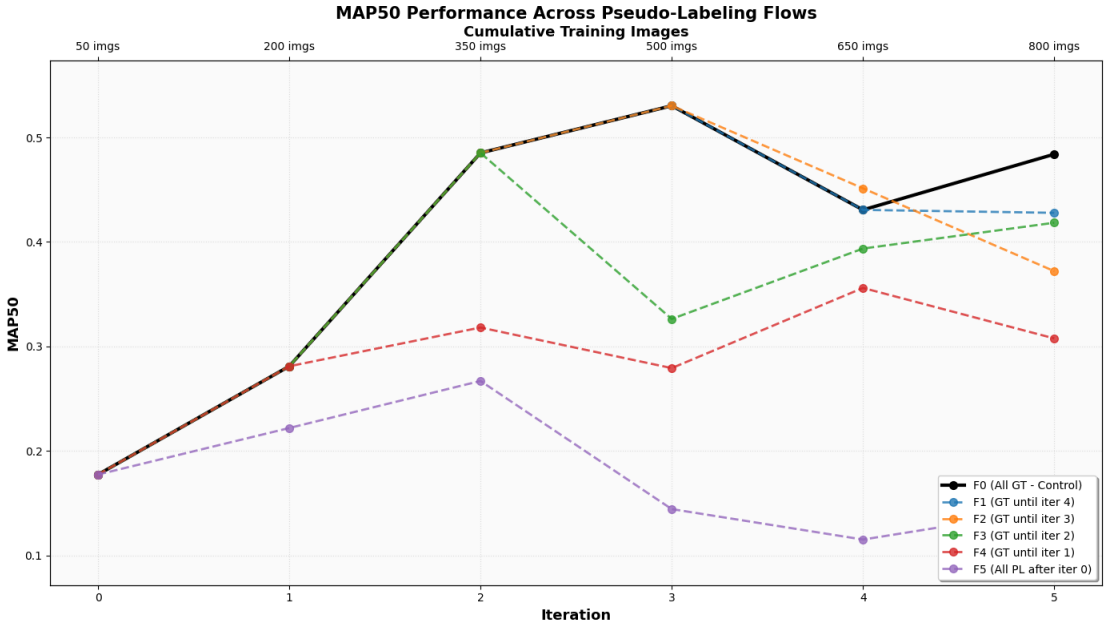


Figure 5.2: mAP@50 Performance Evolution Across All Pseudo-Labeling Flows

Figure 5.2 is a line chart which displays the performance of all flows across five iterations, with the x-axis representing iteration numbers and cumulative training data sizes and the y-axis representing the mAP@50. All flows begin at the common baseline of mAP@50 = 0.177 with 50 training images. This graph is based on the complete experimental data as seen in Table A.1 in the appendix. Each line represents a different experimental flow, where the solid black line (F0) shows the performance trajectory when using only ground truth annotations throughout all iterations. Since this flow uses only ground truths when expanding the training set, it serves as the control baseline. The dashed lines (F1-F5) represent flows that introduce pseudo-labeling at different stages, with each dashed line diverging from the solid F0 line at the specific iteration where pseudo-labeling begins. For instance, F1 follows F0 identically until iteration 4, where it diverges as pseudo-labels are introduced for the first time. Up until those pseudo labels are introduced, the training set is the exact same as the one used in F0. Similarly, F2 diverges at iteration 3, F3 at iteration 2, F4 at iteration 1, and F5 immediately after the initial model setup. This divergence pattern follows the experimental setup explained in Figure 5.2. From the graph it can be seen that flows with higher ground truth proportions (F0, F1, F2) steadily improve until iteration 3 where the peak performance reaches mAP@50 = 0.53. Across each flow (besides flow 0) it is clear that the introduction of pseudo-labeling creates points in the graph where performance almost always declines by varying amounts. To understand these patterns more clearly, each flow should be examined individually:

1. **F0 (Full Ground Truth)**: This flow is intended to be the control baseline and achieved the highest performance across all metrics with mAP@50 of 0.484 and mAP@75 of 0.462. The performance shows consistent improvement from the initial baseline of 0.177 until reaching peak performance at iteration 3 (mAP@50: 0.530). However, after reaching this peak of 0.530, it is observed that there is a decline, with performance dropping to 0.484 by iteration 5. This decline pattern is seen across all flows. Despite this issue, F0 remains the strongest overall acting as an upper bound.

2. **F1 (4 GT + 1 PL)**: This flow was highly competitive with mAP@50 of 0.428, which is only a 11.6%

decrease compared to the control. F1 diverges at iteration 4 when 150 pseudo labeled images are added, resulting in a performance drop from 0.484 to 0.428. This drop in performance is the smallest across all flows that introduced pseudo-labels. Although it might seem logical to infer that 650 GT images provides an optimal foundation before adding pseudo labels, other flows in the experiment show different patterns that challenge this assumption, making it difficult to establish a direct relationship between GT quantity and pseudo-labeling success.

3. **F2 (3 GT + 2 PL)**: This flow achieved moderate performance with mAP@50 of 0.372, which is a 23.1% decrease compared to the control. F2 diverges at iteration 3 when pseudo-labels are first added. The introduction of pseudo labels in this case resulted in a decline from 0.451 to 0.372 over the final two iterations. The performance drop is more significant than the one in F1.

4. **F3 (2 GT + 3 PL)**: This flow achieved mAP@50 of 0.418, which is a 13.6% decrease compared to the control. F3 diverges at iteration 2 when pseudo-labels are introduced and performance initially drops to 0.326 but then recovers and increases back up to 0.418 by iteration 5. This flow had the most substantial recovery performance among all the flows.

5. **F4 (1 GT + 4 PL)**: This flow showed weak performance with a final mAP@50 of 0.308 (36.4% decrease from the control). F4 starts with only 200 ground truths and after the introduction of pseudo labels, it is clear that the performance doesn't increase substantially.

6. **F5 (All PL)**: Flow 5 relied solely on the pseudo labels across each iteration and suffered the most with regards to performance (mAP@50 = 0.144, 70.2% decrease compared to control). This model strictly relies on the capabilities of the initial model and while there is a slight improvement through iteration 2 (mAP@50 = 0.267), the performance ends below the performance of the initial model. After a thorough inspection of the results and predictions, the lack of ground truth data led to unchecked error propagation in which the model failed to correct itself after each iteration. This resulted in the model reinforcing its own errors and degrading in performance compared to the other flows.

Precision-Recall Trade-offs

Precision and recall are important metrics to look at when evaluating the performance of models. Precision measures the proportion of predicted positive instances that are actually correct. High precision means fewer false positives (lower probability of classifying a false instance as true).

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{5.2}$$

Recall measures the proportion of actual positive instances that were correctly identified. Higher recall means fewer false negatives (lower probability of classifying a true instance as false).

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5.3}$$

The precision-recall trade-off is a key challenge in machine learning where improving precision (reducing false positives) often results in degrading recall (increasing false negatives), and vice versa. All model evaluations used a confidence threshold of 0.05, meaning only predictions with confidence scores above 5% were considered as positive detections during testing. This very low evaluation threshold ensures high recall by capturing most potential detections, though it may include some false positives that affect precision. It's important to note that generated pseudo-labels used a much higher confidence threshold of 0.9, meaning only predictions with greater than 90% confidence were converted into training pseudo-labels. This high threshold for pseudo-label generation ensures that only high-quality, confident predictions are added to the training set. This threshold was not arbitrarily chosen but determined through active analysis of the pseudo-label quality and identified as the optimal balance between confidence and dataset expansion needs. In the context of the AVL Asparagus Dataset and the goal of the harvesting project, this trade-off has direct consequences with regards to operations. Missing a detection (low recall) means the harvesting machine fails to harvest a viable asparagus spear, resulting in lost yield. False detections (low precision) results in causing unnecessary machine movements and potential damage to the field or equipment. In this case since both false positives and false negatives are equally important, the goal is to find a balance between precision and recall and ideally have both values high and close to each other for the success of deployment.

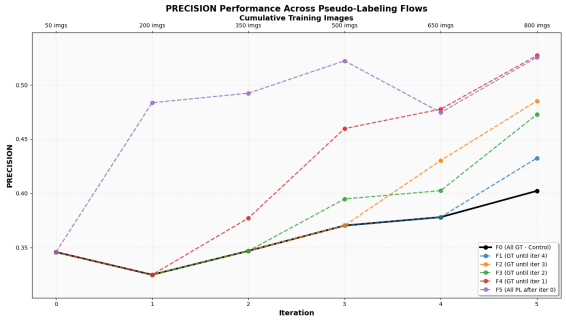


Figure 5.3: Precision Performance Across Pseudo-Labeling Flows

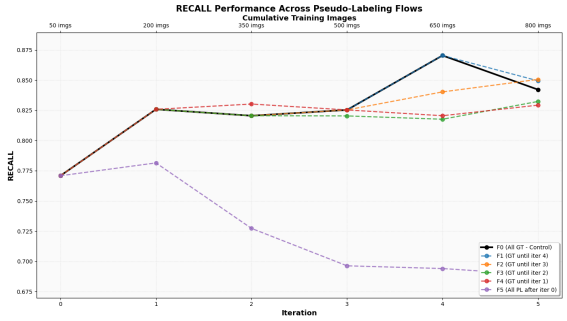


Figure 5.4: Recall Performance Across Pseudo-Labeling Flows

Figures 5.3 and 5.4 display the precision and recall performance of all flows across five iterations, with the x-axis representing iteration numbers and cumulative training data sizes. Both graphs follow the same experimental structure as the mAP@50 analysis line chart. The control flow (F0) demonstrates the most pronounced precision-recall trade-off, with precision varying from 0.325 to 0.402 across iterations while recall remains between 0.821 and 0.870. In contrast, flows with more pseudo-labeling (F2, F3, F4) achieved higher precision scores while maintaining competitive recall rates. This suggests that while pseudo-labels may miss some instances (affecting recall), the labels they do generate tend to be more conservative and accurate.
A deeper analysis shows that F4 achieved the best trade-off between these metrics, with the smallest difference of 0.302 between precision (0.527) and recall (0.829). This was calculated by taking the absolute difference between precision and recall scores for each flow, where smaller differences indicate better balance. The control flow (F0) had the largest gap of 0.440. Although F4 had low mAP in compared to other flows, its trade-off between recall and precision is the best as it has the most balanced relationship between the two metrics, which could be valuable in scenarios where false positives and false negatives are equally important. Despite having the second-lowest mAP scores, F4 achieved the highest precision (0.527) and maintained competitive recall (0.829), resulting in the highest F1-score (0.644) among all flows as seen in Figure A.2 in the appendix. The F1-score represents the mean of precision and recall which means its a single metric that balances both measures.

Overall Efficiency and Optimal Threshold

To determine the most practical pseudo-labeling approach for real-world deployment, it is essential to evaluate the trade-off between annotation effort and model performance. This analysis aims to identify the optimal balance point where minimal manual supervision still produces viable model performance for agricultural applications. The efficiency ratio takes into account the actual performance gain achieved per unit of manual annotation effort.

Table 5.3: Training Data Composition and Performance Efficiency

Flow	Manual Effort (%)	Performance Gain (mAP@50)	Efficiency Ratio
F0	100%	0.307 (0.484 - 0.177)	3.07
F1	81.25%	0.251 (0.428 - 0.177)	3.09
F2	62.5%	0.195 (0.372 - 0.177)	3.12
F3	43.75%	0.241 (0.418 - 0.177)	5.51
F4	25%	0.131 (0.308 - 0.177)	5.24
F5	6.25%	-0.033 (0.144 - 0.177)	-5.28

Efficiency Ratio = (Flow's Final mAP@50 - Initial Model mAP@50) / Manual Annotation Effort (%)

The efficiency ratio measures the actual performance gain achieved per unit of manual effort where higher positive values indicate better efficiency. This metric uses the manual annotation effort as a parameter and provides a more accurate way of assessing performance because it accounts for the baseline performance and penalizes flows that perform worse than the initial model. F3 has the highest efficiency ratio of 5.51 which means it achieves 5.51 mAP points of improvement for every percentage point of manual effort invested. F4 follows closely with an efficiency ratio of 5.24 and demonstrates that minimal ground truth foundations can still achieve reasonable efficiency when combined with extensive pseudo-labeling. The control flows (F0, F1,

F2) show similar efficiency ratios around 3.0-3.1, which is good but is impacted by the high manual effort. Most importantly, F5 exhibits a negative efficiency ratio of -5.28 which clearly demonstrates that relying solely on pseudo-labels without sufficient ground truth foundation actually degrades performance below the baseline and represents a counterproductive approach. This analysis confirms F3 as the optimal balance point for cost-effective pseudo-labeling deployment since it maximizes performance gains while minimizing manual annotation requirements.

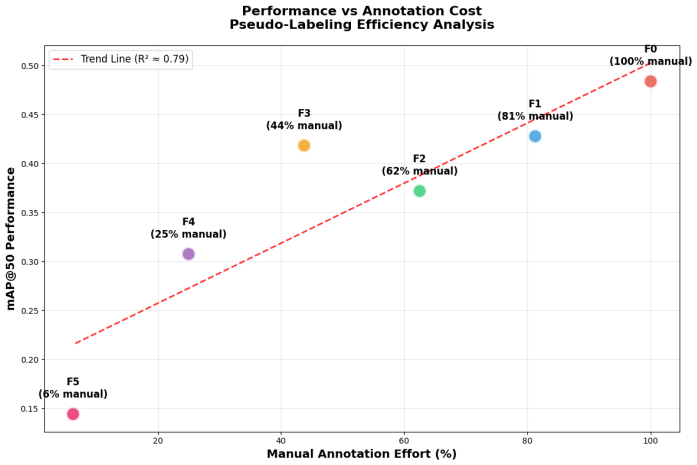


Figure 5.5: Performance vs Annotation Cost - Pseudo-Labeling Efficiency Analysis

Figure 5.5 illustrates the relationship between manual annotation effort and model performance across all experimental flows. The x-axis represents the percentage of manual annotation effort required, while the y-axis shows the final mAP@50 performance achieved. Each point represents a different flow, with the trend line ($R^2 = 0.79$) demonstrating a strong positive correlation between manual effort and performance. However, the graph reveals that F3 (44% manual effort) significantly outperforms the expected trend line, achieving 86.4% of baseline performance while requiring less than half the annotation effort. This visualization confirms F3 as the optimal balance point for cost-effective pseudo-labeling deployment, as it provides the best performance-to-cost ratio while remaining above the efficiency trend line. F5 (6% manual) falls well below the trend line, indicating insufficient manual supervision for viable model performance. It is important to keep in mind that this efficiency ratio is solely based on mAP@50 without considering other metrics like recall and precision.

5.4.3 Economic Impact Assessment

To evaluate how practical pseudo-labeling is for industrial deployment, it is essential to dive into the economic benefits achieved through reduced manual annotation requirements. Based on current annotation service rates for the AVL Asparagus dataset, each image costs €0.174 on average to annotate manually. Using this price point, it is possible to convert the efficiency ratios into concrete cost reductions and evaluate the potential of each pseudo-labeling strategy for proof-of-concept scenarios with tight deadlines.

Table 5.4: Economic Analysis of Pseudo-Labeling Flows

Flow	GT Images	Annotation Cost	mAP@50	Cost per mAP Point	Cost Reduction
F0	800	€139.20	0.484	€287.60	0% (baseline)
F1	650	€113.10	0.428	€264.25	18.7%
F2	500	€87.00	0.372	€233.87	37.5%
F3	350	€60.90	0.418	€145.69	56.3%
F4	200	€34.80	0.308	€112.99	75.0%
F5	50	€8.70	0.144	€60.42	93.8%

The table above (5.4) shows the different costs and final performance associated with each flow which has different amounts of ground truth data.

Manual Correction Workflow Analysis

To evaluate the practical efficiency of the manual correction phase within the pseudo-labeling pipeline, a separate workflow analysis was conducted using the AVL Asparagus dataset. Using the same initial model used in the previous experiment, pseudo-labels were generated for additional batches of 50 images across five iterations. Each batch of pseudo-labeled predictions were then manually corrected to ground truth standards and specific data was collected. This analysis helps investigate how the correction workload decreases as the underlying model improves through successive iterations.

Table 5.5: Manual Correction Effort Analysis Across Iterations

Iteration	Corrections Made	Correction Time	Corrections/Image
1	47	00:24:00	0.94
2	36	00:23:00	0.72
3	32	00:16:07	0.64
4	27	00:16:12	0.54
5	20	00:15:00	0.40

The correction workflow reveals a clear improvement pattern across every iteration. The number of corrections required decreased from 47 in the first iteration to 20 in the fifth iteration, representing a 57.4% reduction in correction effort. Similarly, the corrections per image metric improved from 0.94 to 0.40. This performance indicates that each successive model iteration was able to produce more accurate pseudo-labels which required fewer manual adjustments. The correction time also decreased from 24 minutes to 15 minutes, showing improved annotation efficiency as the model learned from previous corrections. Its important to note that the decrease in the number of corrections made could also be influenced by the fact that the person annotating simply got better and faster at annotating the images.

5.5 Discussion

5.5.1 Ground Truth Threshold and Model Performance (Sub-Question 2)

F3 demonstrated that 350 ground truth images were a sufficient foundation for effective pseudo-labeling. That flow achieved 86.4% of baseline performance while requiring only 43.75% manual annotation effort and also had the most substantial recovery performance among all flows, which suggests that starting with 350 GT images might be the most sufficient foundation for the model to adapt to pseudo-labels over time. F4 achieved the best precision-recall trade-off with the smallest difference of 0.302 between precision (0.527) and recall (0.829), resulting in the highest F1-score (0.644) among all flows which is much better and higher than the precision-recall trade-off of the control F0. This balance is particularly valuable for asparagus harvesting where both false positives and false negatives carry equal operational costs.

The performance decline seen in F0 provides clear evidence of possible overfitting to the training data. By conducting a deeper analysis on the values of F0, we can see that while mAP@50 peaks at iteration 3 (0.530) and then declines to 0.484 by iteration 5, the precision-recall metrics exhibit concerning instability. F0's precision fluctuates from 1 through 5 which demonstrates the model's inability to maintain consistent decision boundaries as training data increases. Despite having access to 800 ground truth images, F0 achieves a precision-recall gap of 0.440 (0.842 recall vs 0.402 precision) which indicates the model has learned to over-predict positive instances. This potential overfitting effect is further impacted due to the dataset's significant class imbalance where some classes contain over 12,000 instances while others have fewer than 1,000. Random sampling at each iteration when expanding the dataset likely magnifies this imbalance and forces the model to memorize patterns rather than learning to generalize. The reasoning for this decline could be due to many different factors including overfitting or class imbalance issues, but the data suggests that the model is fitting to noise in the training data.

Flows incorporating pseudo-labels demonstrate much better precision-recall balance which provides evidence that having noise helps regularize the model. F4, despite using only 200 ground truth images, achieves the best precision-recall balance with a gap of only 0.302 (0.527 precision vs 0.829 recall) and the highest F1-score (0.644) across all flows. This improvement aligns with established research showing that pseudo-labeling can

act as a regularization technique. Xie et al. [24] demonstrated in their "Self-training with Noisy Student" framework that adding noise during training consistently improves generalization and reduces overfitting. In addition to this paper, Arazo et al. [25] also showed that pseudo-labels can break confirmation bias in overfitted models by forcing the network to learn more robust feature representations. The much better recall and precision performance in pseudo-labeling flows suggests that the noise introduced through pseudo-labels prevents the model from memorizing specific annotation patterns and instead encourages it to generalize more. This effect seems valuable in agricultural datasets where class imbalance and limited training data often lead to overfitting issues. Since this is a consistent pattern across all training flows, we can infer that simply adding more annotated images doesn't always lead to better performance with regards to this use case and that pseudo-labeling provides a mechanism to achieve better generalization with fewer manual annotations.

5.5.2 Economic and Practical Efficiency (Sub-Question 3)

F3, with 450 pseudo labeled images in the training set and only 350 ground truth images, had the best cost-performance balance at €60.90 compared to F0's €139.20, representing a 56.3% cost reduction while maintaining viable performance. The economic analysis reveals that F3 achieves a cost per mAP point of €145.69 compared to F0's €287.60, demonstrating nearly half the cost efficiency of the baseline approach. Despite not being as efficient-looking as expected (in terms of how much is actually being saved), the table (5.4) demonstrates that pseudo-labeling becomes increasingly more viable as dataset sizes and time constraints increase, with cost reductions ranging from 18.7% (F1) to 93.8% (F5) depending on the supervision strategy employed. Of course, the more annotated data the better, however for rapid model development under tight deadlines, this process is far more cost and time effective. When considering the Asparagus dataset's full scope of 80,000 images at €0.174 per image (approximately €13,920 total), the cost savings become a lot more substantial when a specific level of performance is required. Additionally, in cases where we have more data and can train even heavier models, annotation costs increase substantially as the amount of data being used increases. Using pseudo-labels within the proof-of-concept phase enables rapid model development under tight deadlines while controlling annotation budgets, making it particularly valuable for startup environments and client demonstrations where both speed and cost control are critical.

Manual correction effort decreases by 57.4% across iterations which also demonstrates the self-improving nature of the pipeline, which translates to direct labor cost reductions as fewer corrections are needed over time. This bootstrap effect demonstrates that manual correction creates a positive feedback loop where each corrected batch improves the model's ability to generate more accurate pseudo-labels in subsequent iterations, reducing future annotation expenses. The developed pseudo-labeling pipeline can use this bootstrap effect to expand the initial ground truth dataset in early iterations then transitioning to automated pseudo-labeling as model performance stabilizes, eliminating ongoing annotation costs entirely. This approach allows practitioners to build substantial ground truth foundations during the initial phases when correction effort is manageable, then switch to fully automated pseudo-labeling once the model is good enough, maximizing return on annotation investment. Given the results achieved even with basic baseline configurations, this bootstrapping strategy could lead to significant performance improvements with bigger datasets while maintaining cost efficiency. Especially for proof-of-concept scenarios, this workflow allows for consistent increase in annotated data while maintaining high annotation quality and controlling costs which is necessary for agricultural deployment. The findings establish that pseudo-labeling works effectively for agricultural object detection applications, especially when the process is conducted across multiple iterations with a healthy amount of ground truth initial data while achieving substantial cost savings.

Chapter 6

Use Case 2: Instance Segmentation

6.1 Overview - Cucumber Dataset

The second use case is based on the VDL Smart Trim project, which was developed by VBTI to support automation tasks within greenhouse cucumber production. The primary goal of the system was to enable a robotic platform to accurately identify and segment key aspects of a cucumber plant in order to automate the process of leaf snipping. In greenhouse cucumber fields, one of the main manual tasks is associated with pruning which is the process of removing excess leaves and stems to promote healthy growth and improve harvesting output. This task accounts for 35% of the manual labor involved in greenhouse operations and requires precise detection of nodes and leaf stems to ensure that the robot does not damage the plants. In addition to the pruning, the detections and system as a whole also acts as an inspection tool that allows the robot to assess plant health during the process. In this project, VBTI was responsible for developing the vision system that creates segmentation masks for all relevant aspects of each plant.

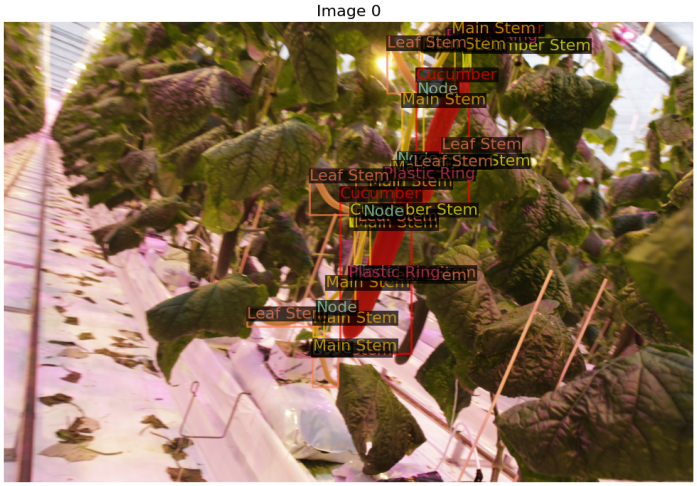


Figure 6.1: Image from the VDL Cucumber dataset

The dataset consists of 2,429 high-resolution greenhouse images and contains the classes mentioned previously: *Cucumber*, *Main Stem*, *Cucumber Stem*, *Leaf Stem*, *Node*, *Plastic Ring*, and *Metal Ring*. On average, each image contains around 70 annotated instances, which makes this dataset substantially more dense than the asparagus use case. All images in the dataset were captured under varying lighting and visibility conditions with different resolutions which makes the dataset even more difficult to work with. The images were collected using a mounted camera system and clearly display variations in stem structure, cucumber sizes, and the number of nodes. As a result, the cucumber dataset is highly relevant for testing the proposed pseudo-labeling framework in environments where cost, accuracy, and label density are main factors. Since pseudo-labeling has not yet been evaluated using niche datasets, the results from this use case are expected to provide valuable insights into the viability and performance of the framework under real world conditions.

6.2 Test Set

Similar to Use Case 1, a test set was created comprising of 150 images with the purpose of providing an unbiased evaluation metric for comparing different models over each iteration. The set was selected to be representative of the overall dataset while remaining completely separate from all training data used in the experiments. The 150 test images contain annotated instances distributed across all seven classes: *Cucumber*, *Main Stem*, *Cucumber Stem*, *Leaf Stem*, *Node*, *Plastic Ring*, and *Metal Ring*. By using this set, it is possible to extract consistent mAP@50, mAP@75, precision, recall, and F1-score metrics across all experimental flows in order to compare the effectiveness of pseudo-labeling within the segmentation framework.

6.3 Chosen Model Architecture

The goal of the VDL Cucumber project was to generate accurate instance segmentation masks for different plant components within a complex greenhouse environment. Given this objective, the model chosen for this task is **Mask R-CNN** [26], as it is specifically designed to perform object detection and instance segmentation simultaneously. This model builds on the Faster R-CNN architecture by introducing an additional branch that predicts a binary segmentation mask for each region of interest which enables instance segmentation alongside object classification. Mask R-CNN follows a two-stage architecture. In the first stage, a Region Proposal Network (RPN) generates Regions of Interest (RoIs), which are bounding boxes likely to contain object instances. In the second stage, these RoIs are passed through classification and bounding box regression heads to predict the class labels and refine the coordinates of each region. Finally, in a third parallel branch, the RoIs are passed through a fully convolutional network (FCN) that generates a segmentation mask for each instance based on the aligned region. The total loss function combines classification, bounding box regression, and segmentation components as follows:

$$L = L_{\text{cls}} + L_{\text{bbox}} + L_{\text{mask}} \tag{6.1}$$

where:

- L_{cls} is the classification loss (typically cross-entropy),
- L_{bbox} is the bounding box regression loss (usually smooth L1),
- L_{mask} is the per-pixel binary cross-entropy loss for the segmentation masks.

The ResNet-50 backbone was selected because of its strong ability to perform well within complex visual environments. Based on the ResNet paper introduced by He et al. [27], the backbone is meant to offer detailed representational learning through its residual connections which makes it more effective in training while avoiding issues with degradation. This makes it well suitable for the complex data used within this use case where there is a number of overlapping instances, varying lighting conditions, and 70+ instances per image. The SGD optimizer was also used because of its consistent convergence properties which is needed when dealing with high noise images. Creating dense segmentation instances within datasets as complex as the one being used requires very effective feature extraction and precise mask prediction capabilities. This makes the Mask R-CNN with a ResNet-50 backbone an ideal choice for this segmentation task. In the context of applying the pseudo-labeling pipeline to the cucumber dataset, all iterations of the framework will be trained using this architecture, with consistent hyperparameters to ensure each flow and iteration is fairly compared:

- **Architecture**: Mask R-CNN with ResNet-50 backbone
- **Training epochs**: 50 per iteration
- **Batch size**: 6
- **Optimizer**: Default SGD optimizer
- **Data augmentation**: None (to maintain annotation consistency)
- **Retraining approach**: Complete model retraining from scratch each iteration

6.4 Experimentation Results

6.4.1 Initial Model Setup and Baseline Performance

Similar to Use Case 1, all flows began with an identical initial model which was trained on 50 manually annotated images with the model architecture discussed in the section 6.3. This initial model achieved a baseline mAP@50 of 0.04, recall of 0.01, and precision of 0.17, which is very poor as expected given the limited training data and the complexity of this segmentation task. For reference, all model results can be seen in table A.2 within the appendix. While the initial model's performance is extremely suboptimal and can be improved, it serves as a baseline that demonstrates how much room for improvement there actually is. Assuming that some training configurations are enhanced (increased epochs, optimized learning rates, or advanced loss functions) and more annotated data is used, these baseline results could be substantially improved. Therefore, the findings presented in this section represent a lower bound on pseudo-labeling effectiveness for segmentation tasks on this specific dataset.

6.4.2 Multi-Flow Performance Analysis

The multi-flow experiment, as explained in chapter 4, looks into six different pseudo-labeling strategies (F0–F5) across five iterations, with each flow having different levels of ground truth annotation. Table 6.1 displays the experimental design in detail, where GT represents ground truths (or manually corrected) annotations and PL represents pseudo-labels generated by the model. Note that all experimental flows and iterative training procedures described in this section have been conducted in accordance with the mathematical framework outlined in Section 4.2.

Table 6.1: Flow breakdown for Instance Segmentation using Mask R-CNN.

Flow ID	Iteration Breakdown	Description	Model Type
F0-S	[GT, GT, GT, GT, GT]	Fully manual masks (segmentation baseline)	Mask R-CNN (Instance Segmentation)
F1-S	[GT, GT, GT, GT, PL]	4 manual iterations, final pseudo-labels	Mask R-CNN (Instance Segmentation)
F2-S	[GT, GT, GT, PL, PL]	3 manual, 2 pseudo masks	Mask R-CNN (Instance Segmentation)
F3-S	[GT, GT, PL, PL, PL]	2 manual masks, 3 pseudo iterations	Mask R-CNN (Instance Segmentation)
F4-S	[GT, PL, PL, PL, PL]	1 manual, rest pseudo masks	Mask R-CNN (Instance Segmentation)
F5-S	[PL, PL, PL, PL, PL]	All pseudo-labels only (no manual)	Mask R-CNN (Instance Segmentation)

After conducting all experimental flows, Table 6.2 presents the performance metrics for all flows at their final iteration (Iteration 5). This table provides a direct comparison of how different supervision strategies impact segmentation model performance.

Table 6.2: Final Performance Comparison Across All Flows (Iteration 5) - Instance Segmentation

Flow	GT	PL	mAP @50	mAP @75	mAP(All)	Prec.	Recall	F1
F0-S	800	0	0.400	0.200	0.210	0.466	0.396	0.428
F1-S	650	150	0.410	0.196	0.214	0.520	0.390	0.446
F2-S	500	300	0.398	0.200	0.211	0.587	0.362	0.448
F3-S	350	450	0.376	0.176	0.194	0.607	0.333	0.430
F4-S	200	600	0.402	0.189	0.208	0.633	0.325	0.429
F5-S	50	750	0.312	0.089	0.133	0.639	0.190	0.293

Performance Interpretation

As stated previously, all flows began with an identical initial model trained on 50 manually annotated images with mAP@50 of 0.04. The results from Table 6.2 reveal a clear relationship between the amount of ground truth data and model stability. The results show a surprising relationship between the amount of ground truth data and model performance stability. All flows show consistent performance improvement despite number of pseudo-labels used.

The relationship can be seen properly in Figure 6.2:

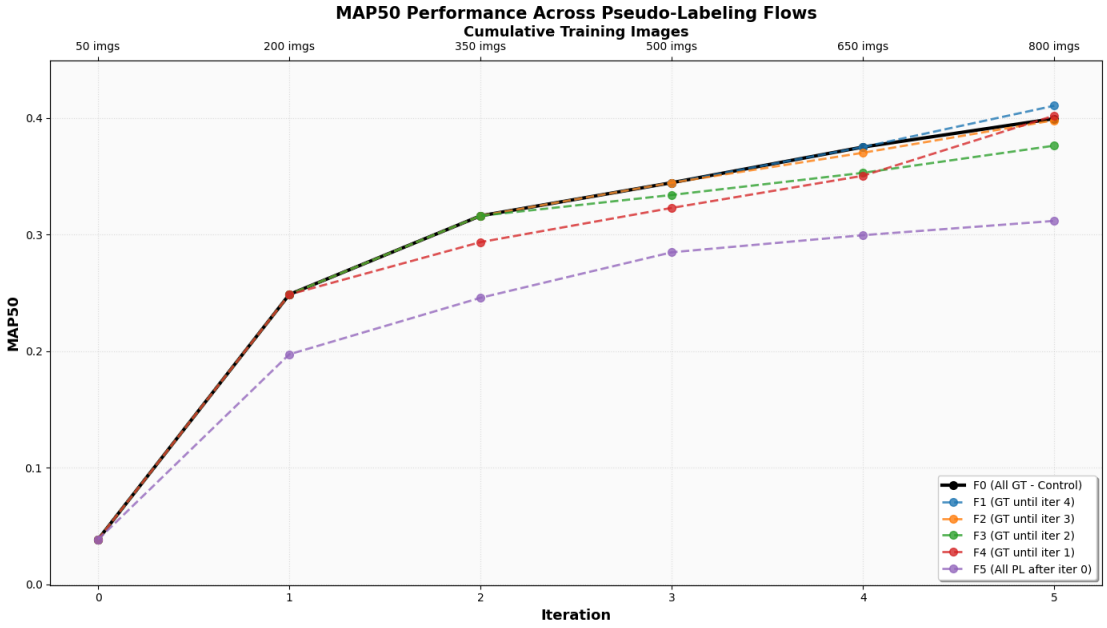


Figure 6.2: Use Case 2 - mAP@50 Performance Evolution Across All Pseudo-Labeling Flows

Figure 6.2 displays the performance evolution of all segmentation flows across five iterations, with the x-axis representing iteration numbers and cumulative training data sizes and the y-axis representing mAP@50. All flows begin at the common baseline of mAP@50 = 0.04 with 50 training images. This graph is based on the complete experimental data as seen in Table A.2 in the appendix. Each line represents a different experimental flow, where the solid black line (F0-S) shows the performance trajectory when using only ground truth annotations throughout all iterations. The dashed lines (F1-S through F5-S) represent flows that introduce pseudo-labeling at different stages and follow the same divergence patters as explained in Use Case 1. Unlike Use Case 1, the segmentation flows show much more consistent performance improvements across all iterations without the declining patterns seen in object detection. To understand these patterns more clearly, each flow should be examined individually:

1. **F0-S (Full Ground Truth)**: This flow serves as the segmentation baseline and achieved the highest performance with mAP@50 of 0.400 and mAP@75 of 0.200. The performance shows consistent improvement from the initial baseline of 0.038 across all iterations: 0.249 → 0.316 → 0.344 → 0.375 → 0.400. Unlike Use Case 1, F0-S demonstrates steady improvement without the performance decline observed in object detection.

2. **F1-S (4 GT + 1 PL)**: This flow achieved the highest performance across all flows with mAP@50 of 0.410, which is a 2.5% increase compared to the control. F1-S follows F0-S identically until iteration 4, where the addition of 150 pseudo labeled images results in a performance increase from 0.375 to 0.410. This flow is one of only two flows which has performance higher than the control after introducing pseudo-labels.

3. **F2-S (3 GT + 2 PL)**: This flow achieved competitive performance with mAP@50 of 0.398, which is only a 0.5% decrease compared to the control. F2-S diverges at iteration 3 when pseudo-labels are first added. The introduction of pseudo labels in this case resulted in minimal performance impact, maintaining near-baseline performance through the final iterations.

4. **F3-S (2 GT + 3 PL)**: This flow achieved mAP@50 of 0.376, which is a 6.0% decrease compared to the control. F3-S diverges at iteration 2 when pseudo-labels are introduced and shows steady improvement across subsequent iterations. This flow demonstrates consistent adaptation to pseudo-labeled data even with limited ground truth foundation.

5. **F4-S (1 GT + 4 PL)**: This flow achieved mAP@50 of 0.402, marginally outperforming the control by 0.5%. F4-S starts with only 200 ground truths and after the introduction of pseudo labels, the performance

increases substantially across each iteration. The extensive use of pseudo-labels combined with minimal ground truth data achieves competitive performance in segmentation tasks.

6. **F5-S (All PL)**: Flow 5 relied solely on pseudo labels across each iteration and achieved mAP@50 of 0.312, representing a 22.0% decrease compared to control. This model strictly relies on the capabilities of the initial model and while there is gradual improvement through iteration 3, the performance plateaus afterwards. The model still demonstrates substantial improvement from the initial baseline, reaching nearly 8x the starting performance.

Precision-Recall Trade-offs

Precision and recall are important metrics to look at when evaluating the performance of models. All model evaluations used a confidence threshold of 0.05, meaning only predictions with confidence scores above 5% were considered as positive detections during testing. It's important to note that generated pseudo-labels used a much higher confidence threshold of 0.8, meaning only pseudo-labels with greater than 80% confidence were converted into training pseudo-labels. This threshold was not arbitrarily chosen but determined through active analysis of pseudo-label quality and was identified as the optimal balance that reserves correct predictions on average per sample that was added. In the context of the VDL Cucumber Dataset and the goal of the automation project, this trade-off has direct consequences with regards to operations. Missing a detection (low recall) means the robotic system fails to identify critical plant components that need to be cleared. False detections (low precision) result in unnecessary robotic movements that can potential damage plant parts. From these results, several flows incorporating pseudo-labels demonstrate performance that matches or even exceeds the fully supervised control flow F0-S, with F1-S and F4-S both outperforming the baseline.

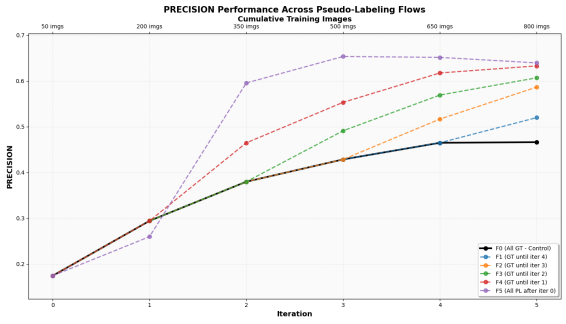


Figure 6.3: Precision Performance Across Pseudo-Labeling Flows - Segmentation

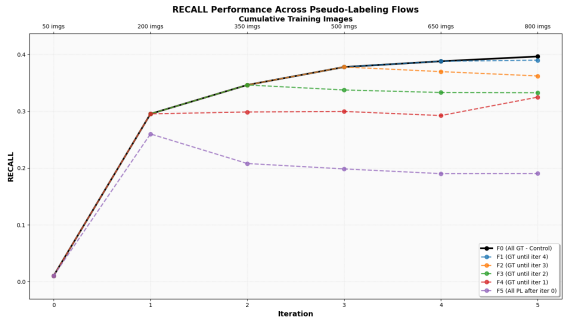


Figure 6.4: Recall Performance Across Pseudo-Labeling Flows - Segmentation

Figures 6.3 and 6.4 display the precision and recall performance of all flows across five iterations, with the x-axis representing iteration numbers and cumulative training data sizes. Both graphs follow the same experimental structure as the mAP@50 analysis line chart. The control flow (F0-S) is seen to have steady precision improvement from 0.294 to 0.466 across each iteration while recall has a big jump after the first iteration (0.01-0.294) then remains relatively stable between 0.295 and 0.396. In contrast, flows with more pseudo-labeling (F2-S, F3-S, F4-S, F5-S) achieved significantly higher precision scores but with lower recall rates. This suggests that pseudo-labels in segmentation tasks tend to be more conservative and accurate but may miss some instances.

A deeper analysis shows that F4-S achieved the highest precision (0.633) among all flows while maintaining reasonable recall (0.325), resulting in a precision-recall gap of 0.308. The control flow (F0-S) had a smaller gap of 0.070 meaning better balance between the two metrics. Although F4-S had competitive mAP compared to the control, its precision-recall trade-off shows the segmentation model's tendency to prioritize accuracy over completeness when trained with extensive pseudo-labels. F1-S achieved the highest F1-score (0.446) among all flows, representing the best overall balance between precision and recall in segmentation tasks. From these results, many of the flows with pseudo labels demonstrate performance that are on par and even exceed the performance of the control flow F0.

Overall Efficiency and Optimal Threshold

To determine the optimal threshold for pseudo-labeling within complex segmentation tasks, the trade-off between performance and manual effort needs to be calculated similar to how it was done in Use Case 1 (5.4). This analysis aims to identify the optimal balance point where minimal manual supervision still produces viable model performance for agricultural segmentation. The efficiency ratio takes into account the actual performance gain per unit of manual annotation effort.

Table 6.3: Training Data Composition and Performance Efficiency - Segmentation

Flow	Manual Effort (%)	Performance Gain (mAP@50)	Efficiency Ratio
F0-S	100%	0.362 (0.400 - 0.038)	3.62
F1-S	81.25%	0.372 (0.410 - 0.038)	4.58
F2-S	62.5%	0.360 (0.398 - 0.038)	5.76
F3-S	43.75%	0.338 (0.376 - 0.038)	7.73
F4-S	25%	0.364 (0.402 - 0.038)	14.56
F5-S	6.25%	0.274 (0.312 - 0.038)	43.84

Efficiency Ratio = (Flow's Final mAP@50 - Initial Model mAP@50) / Manual Annotation Effort (%)

The efficiency ratio measures the performance gain achieved per unit of manual effort where higher positive values indicate better efficiency with regards to how much ground truth data was used. This metric takes into account the performance growth since the initial model and highlights the true value gained from manual annotation investment. F0-S demonstrates the baseline efficiency ratio of 3.62, which is relatively low but understandable due to the high level of manual effort. All flows with pseudo labels achieved very competitive levels of performance compared to the control and from the table it can be seen that as manual effort decreases across the flows, efficiency ratios consistently increase with F1-S achieving 4.58, F2-S reaching 5.76, and F3-S attaining 7.73, showing that moderate pseudo-labeling strategies can achieve significant efficiency gains compared to full supervision. F5-S exhibits the highest efficiency ratio of 43.84 which indicates exceptional efficiency with regards to manual efforts, however its performance (0.312 mAP@50) seems insufficient for deployment when compared to the performance of other flows. F4-S emerges as the optimal choice with an efficiency ratio of 14.56, achieving competitive performance (0.402 mAP@50) while requiring only 25% manual annotation effort. The segmentation flows show consistently higher efficiency ratios compared to object detection (ranging from 3.62 to 43.84 versus 3.07 to 5.51) which highlights the effectiveness of pseudo-labeling for dense annotation tasks like instance segmentation where the cost and complexity of manual annotation are substantially higher. This efficiency advantage is seen across all flows with high to low amounts of pseudo labeling, making it extremely valuable when considering that segmentation masks require pixel-level precision compared to simple bounding boxes.

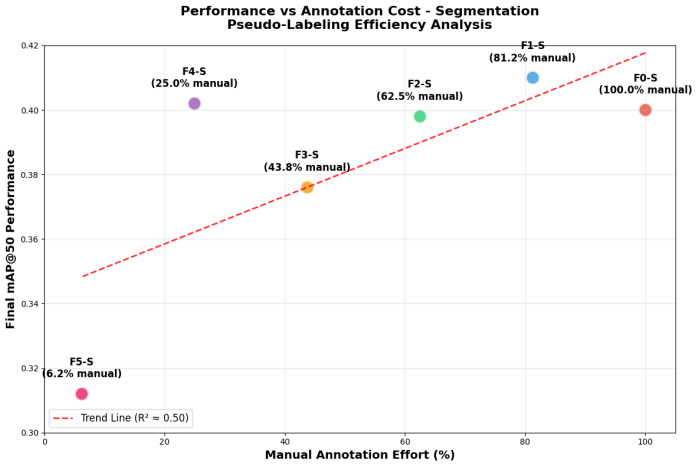


Figure 6.5: Performance vs Annotation Cost - Pseudo-Labeling Efficiency Analysis (Segmentation)

Figure 6.5 illustrates the relationship between manual annotation effort and segmentation model performance

across all experimental flows. The visualization demonstrates that segmentation tasks exhibit different efficiency dynamics compared to object detection and clearly shows the effectiveness of F4-S in achieving the most optimal cost-effectiveness by maintaining near-baseline performance while requiring only 25% manual effort. The trend analysis reveals that using pseudo labeling within the cucumber segmentation dataset provides more consistent efficiency gains across each supervision level, making it more suitable for deployment compared to the object detection use case.

6.4.3 Economic Impact Assessment

To evaluate how practical pseudo-labeling is for segmentation tasks in industrial deployment, it is essential to dive into the economic benefits achieved through reduced manual annotation requirements. Based on current annotation service rates for the VDL Cucumber dataset, segmentation mask annotations cost significantly more than bounding boxes due to being more complex to annotate. Individual annotation prices range from €0.0555 to €0.1725 depending on the class, with an average cost of approximately €0.11 per instance. Given that each image contains an average of 70 annotated instances, the cost per image is substantially higher than object detection tasks. The instance distribution for the cucumber dataset can be seen in the instance distribution graph in the appendix (A.4).

Table 6.4: Economic Analysis of Pseudo-Labeling Flows - Instance Segmentation

Flow	GT Images	Annotation Cost	mAP@50	Cost per mAP Point	Cost Reduction
F0-S	800	€6,160.00	0.400	€15,400.00	0% (baseline)
F1-S	650	€5,005.00	0.410	€12,207.32	18.7%
F2-S	500	€3,850.00	0.398	€9,673.37	37.5%
F3-S	350	€2,695.00	0.376	€7,167.55	56.3%
F4-S	200	€1,540.00	0.402	€3,831.34	75.0%
F5-S	50	€385.00	0.312	€1,233.97	93.8%

The table above shows the different costs and final performance associated with each segmentation flow. The economic impact is substantially more significant than object detection due to the higher base annotation costs for segmentation masks. F4-S demonstrates excellent cost efficiency while achieving competitive performance (0.402 mAP@50) at only €1,540 compared to F0-S's €6,160. This represents a 75% cost reduction while maintaining 100.5% performance retention. F5-S has the highest cost saving potential at 93.8% but with performance retention of only 78%. All of the other flows also demonstrate compelling economic advantages over the control baseline. F1-S achieves the highest overall performance (0.410 mAP@50) while providing an 18.7% cost reduction. F2-S offers a balanced middle ground with 37.5% cost savings and competitive performance (0.398 mAP@50), representing only a 0.5% performance decrease from the control. F3-S provides substantial cost reduction at 56.3% while maintaining reasonable performance (0.376 mAP@50). These results demonstrate that segmentation tasks offer multiple viable pseudo-labeling strategies depending on project constraints, with all flows except F5-S maintaining performance within 6% of the baseline while delivering significant cost reductions.

Manual Correction Workflow Analysis

To evaluate the practical efficiency of the manual correction phase within the pseudo-labeling pipeline for segmentation tasks, a limited manual correction analysis was conducted using the VDL Cucumber dataset. Unlike the object detection use case, segmentation corrections were significantly more challenging and time-intensive due to the complex masks required and the high number of instances per image.

Table 6.5: Manual Correction Effort Analysis - Instance Segmentation

Iteration	Corrections Made	Correction Time	Corrections/Image
1	220	02:25:05	4.40

The correction workflow analysis was limited to a single iteration due to the challenges encountered during the manual correction process. The initial model's extremely poor performance (mAP@50 = 0.038) resulted in

pseudo-labels that required extensive corrections, with 220 corrections needed across 50 images, representing 4.40 corrections per image. In addition to the number of corrections required, the correction time was 2 hours and 25 minutes for just 50 images. This demonstrates the significant time investment required for segmentation mask corrections compared to bounding box adjustments. For reference, when outsourced, the annotation provider in context can only output 100-200 annotated images per week. Each correction involved precise pixel-level adjustments to segmentation boundaries, addition of missed instances, and removal of false positive masks, which made the process a lot more labor-intensive than object detection corrections. Given the poor quality of initial pseudo-labels and the long turnaround time, the manual correction analysis was very limited. This limitation shows the importance of having enough ground truth images for to train a baseline model with reasonable performance before implementing manual correction workflows into segmentation tasks.

6.5 Discussion

6.5.1 Ground Truth Threshold and Model Performance (Sub-Question 2)

While Use Case 1 found that F3 achieved the optimal balance with 350 ground truth images for object detection with 86.4% performance retention and 56.3% cost reduction, the segmentation task reveals different optimal thresholds. F4-S shows the most compelling case for optimal efficiency in segmentation tasks with very high performance (mAP@50 of 0.402) and only 200 ground truth images. This flow resulted in 100.5% performance retention with a 75% cost reduction compared to the control flow (F0-S) with 800 ground truth images. F4-S achieved the highest precision (0.633) among all flows while maintaining reasonable recall (0.325), resulting in a precision-recall gap of 0.308 which although is not the most balanced, is compensated for by its high performance and low cost.

Although F4-S performed very well, F1-S was the second flow to outperform the control by a fair margin, using 150 pseudo labels and 650 ground truth images. F1-S demonstrated that the addition of pseudo labels provided an excellent enhancement to the segmentation model, achieving the highest performance with mAP@50 of 0.410 which represents a 2.5% improvement over the baseline. These two flows not only reveal potential for high time and cost savings but also demonstrate that pseudo-labels can actually increase performance when trained on the same dataset size as a fully supervised model. Unlike the papers discussed in the Related Work chapter which all used larger datasets when incorporating pseudo-labels to achieve performance gains, this segmentation framework achieved superior performance without needing additional data. This shows that pseudo-labeling can act not only as a cost-reduction tool but as a primary approach for improving model performance in complex segmentation tasks.

In addition to the successful results seen in this use case, the segmentation flows have different learning dynamics without the overfitting issues observed in object detection (Use Case 1). F0-S shows steady, consistent improvement across all iterations ($0.249 \rightarrow 0.316 \rightarrow 0.344 \rightarrow 0.375 \rightarrow 0.400$) without performance decline which suggests that segmentation models benefit from additional training data without suffering from overfitting issues. This difference can be a result of the inherent complexity of segmentation tasks which require deep representations rather than simple bounding box coordinates. The complexity of the task means that segmentation models generally need more data to reach high performance and the dense annotation nature (70+ instances per image) provides more diverse training signals which prevents the model from memorizing specific patterns.

Similar to the benefits of introducing noise observed in Use Case 1 (5.4), the segmentation flows also reveal that pseudo-labeling provides benefits that actually enhance performance. F1-S and F4-S both outperformed the control baseline which can be an indication to the fact that pseudo-labels can actually capture relationships and boundary details that manual annotations might miss or annotate inconsistently. This is most clearly seen in F1-S where adding just 150 pseudo-labeled images resulted in a 2.5% performance improvement over the fully supervised baseline. F4-S further supports this where extensive pseudo-labeling (600 images) combined with minimal ground truth (200 images) achieved performance comparable to much higher supervision levels which could mean that segmentation models can effectively learn from pseudo-generated mask boundaries once a basic model is established.

6.5.2 Economic and Practical Efficiency (Sub-Question 3)

F4-S achieved the best cost-performance balance in this use case. This flow required only €1,540 compared to F0-S's €6,160 and achieved even better performance (0.402 vs 0.400 mAP@50). When put in perspective, this represents a 75% cost reduction with actually better performance than the control. The economic analysis reveals that F4-S achieves €3,831 per mAP@50 point compared to F0-S's €15,400 per mAP@50 point. This is a 4x improvement in cost efficiency which further signals the great value that pseudo-labeling brings to segmentation workflows. While the cost savings in the object detection use case were meaningful, they were not as dramatic due to the low base annotation costs of €0.174 per image for bounding boxes. The segmentation task's much higher base annotation costs (€7.70 per image vs €0.174 for object detection) amplify the economic benefits of pseudo-labeling dramatically, with cost reductions ranging from 18.7% (F1-S) to 93.8% (F5-S) depending on the flow. When considering that segmentation masks are very complex and require much more time to annotate, the economic advantages become a lot more compelling as the cost savings are not just substantial but can potentially transform the way these models are developed and deployed. The combination of reduced costs and improved performance makes pseudo-labeling particularly attractive for segmentation tasks where annotation expenses can quickly become prohibitive for large-scale projects.

When considering that the full VDL Cucumber dataset is made up of 2,429 images at €7.70 per image (approximately €18,703 total), the cost savings can be seen as even more substantial. Assuming that we scale to use the whole dataset and the performance trends observed in our experiments remain the same, F4-S's approach could potentially reduce annotation costs from €18,703 to approximately €4,676 while maintaining viable performance. This represents savings of over €14,000 for a single dataset. These savings become even more significant when scaled to continuous production environments where new data collection and annotation are ongoing requirements. Beyond the direct cost benefits, the time savings are equally compelling as current annotation providers can only deliver 100-200 annotated segmentation images every two weeks, meaning the full dataset would require approximately 24-48 weeks to complete through traditional manual annotation. The efficiency ratio analysis confirms F4-S as exceptionally efficient with a ratio of 14.56, meaning it achieves 14.56 mAP points of improvement for every percentage point of manual effort invested, compared to the object detection optimal flow (F3) which achieved 5.51.

The time constraints of segmentation annotations create additional economic pressures that make pseudo-labeling very valuable for time-sensitive projects. With annotation providers requiring nearly 2 weeks per 200 images, pseudo-labeling offers a pathway to accelerate deployment timelines while simultaneously reducing costs. This time-to-market advantage becomes crucial for scenarios where delayed deployment can result in lost business opportunities. The combination of reduced annotation time, lower costs, and maintained performance creates a compelling value proposition that extends beyond simple cost optimization to strategic business advantages.

The extremely poor baseline performance (mAP@50 = 0.038) made extensive manual corrections impractical which suggests that segmentation workflows require careful initial model development or alternative bootstrapping strategies to achieve viable starting points. However, once this threshold is reached, segmentation models are seen to offer much scalability potential that justifies the initial investment in establishing proper foundations. The findings establish that pseudo-labeling can be a valuable tool within segmentation projects and can help in moving from traditional annotation-heavy workflows to more semi-supervised strategies that prioritize early solid performance followed by automated scaling using pseudo labels.

Chapter 7

Discussion

7.1 Fully-Supervised vs Pseudo-Labeling Framework Comparison

The research question of the paper involves looking at the benefits that are associated with pseudo-labeling workflows within complex agricultural environments. The focus of this research question aims to evaluate the difference between fully supervised and pseudo-labeling approaches in terms of model performance, cost efficiency, and the overall viability of using these workflows within industrial datasets. After conducting each experiment, the results from both use cases provide clear evidence that pseudo-labeling workflows are capable of offering significant advantages over traditional fully supervised approaches. The results reveal lots of room for cost optimization, a reduced level of outsourcing, and performance patterns that were not initially expected and which exceeded expectations.

7.1.1 Performance and Training Efficiency

The fully supervised control flows (F0 and F0-S) in both use cases had consistently strong results across all performance metrics. In use case 1 however, F0 showed clear signs of overfitting with the first being the inconsistent mAP@50 seen across each iteration (peaking at iteration 3 (mAP@50 = 0.530) and then declining to 0.484 by the final iteration 5). This notable decline along with the precision and recall fluctuations that occurred across each iteration both suggest that the model was essentially memorizing specific patterns rather than generalizing. This decline pattern was observed across all other flows within use case 1, however from the experimental results it was concluded that the pseudo labels, which were added to some flows, actually offered beneficial noise and helped with reducing potential overfitting issues. In use case 2, there were no signs of overfitting and the results were even more compelling since some flows with pseudo labels actually outperformed than the control baseline. From these findings it can be concluded that simply adding more manually annotated data does not always lead to better overall performance and can actually hurt the model's ability to perform well on unseen data.

After thorough evaluation of all the performance metrics collected from both experimental use cases, it can also be clearly seen that pseudo-labeling flows demonstrated consistently better precision-recall balance when compared to the 2 control flows. F4 in the object detection task achieved the best overall trade-off with a precision-recall gap of only 0.302 (compared to F0's gap of 0.440). Since both false positives and false negatives are equally important in the context of the use cases, this more balanced result was even better compared to the control. In segmentation, F1-S and F4-S both outperformed the fully supervised baseline, with F1-S achieving 2.5% better performance (0.410 vs 0.400 mAP@50). This further aligns with some of the literature discussed in the related work section which emphasizes the benefits of noise when introduced within a training set. F4-S on the other hand was the most surprising flow across both use cases as it managed to match the baseline performance of the control with only 25% manual effort. These results show that pseudo-labeling is equally as effective on real-world agricultural datasets (uncontrolled) as the controlled benchmark datasets used by Zou et al. (2020) and Hu et al. (2022). While the existing literature demonstrated performance improvements by adding lots of pseudo-labeled data to small ground truth foundations and using specific filtering techniques, our framework proves that pseudo-labeling can achieve comparable or superior results while maintaining much smaller overall dataset sizes and lower annotation costs.

7.2 Ground Truth Thresholds and Economic Efficiency Summary

From a practical and financial perspective, the workflows tested in this paper were seen to bring lots of advantages with regards to cost reduction and development speed. The segmentation task had the most substantial economic benefits by far as F4-S achieved 75% cost reduction (€1,540 vs €6,160) while still having the same performance as the control flow. Object detection also had a good amount of savings with F3 achieving a 56.3% cost reduction while still retaining 86.4% of the original performance. However, although these numbers look very enthusiastic, this only translates to approximately €78.30 in saved money for the 800 image dataset used in the experiments. The difference in actual cost savings between the two tasks directly shows how much more dense the annotations are within the segmentation task is when compared to the relatively simple process of drawing bounding boxes around objects.

The optimal ground truth thresholds and overall results between the two use cases were seen to differ in the experimentation phase. The object detection flow was observed to require approximately 350 ground truth images for stable and competitive performance, while instance segmentation achieved competitive performance with only 200 ground truth images (F4-S). This finding could be explained by the difference in information density between the two annotation types. Object detection requires only four coordinate points to define a bounding box around an asparagus spear (and has only 3 instances per image on average), while instance segmentation demands precise pixel boundaries to capture the specific details of each cucumber plant and other components. A single segmentation mask for a cucumber contains many pixel classifications that contribute to the model's understanding of object boundaries and context. In object detection, a bounding box provides spatial information of the object location and the approximate size. This difference could be a factor that amplifies this effect of how much ground truth images is required to bootstrap the pseudo labeling flow. This means that 200 segmentation images effectively provide the necessary learning information of potentially 1,000+ object detection images, which helps to explain why segmentation models can achieve strong performance with fewer ground truth samples. The model learns not just to identify objects, but to understand many of the complex factors presented in each image. Despite these differences in optimal thresholds, the analysis conducted throughout this section and the experimentation section of both use cases concludes that both tasks majorly benefit from the pseudo-labeling workflow. The prior research discussed in the Related Work chapter had a different overall focus compared to this paper, as the studies discussed in those chapters focused on performance improvements while the priority within this research was to prioritize cost optimization as the primary objective and demonstrate the effectiveness of pseudo-labeling on niche agricultural applications.

7.3 Limitations and Future Work

Despite the results and experiments within this research being successful, every research project comes with limitations and opportunities for future investigation. Firstly, the experiments conducted within the two use cases were limited to five iterations per flow, which may hinder the true potential of the workflow. Future experiments should extend the number of iteration to 10-15 to better understand performance convergence patterns and optimal stopping criteria. Some flows showed recovery patterns that might continue improving with more iterations, making longer studies valuable and there could potentially be more pseudo flows that outperform flows that are fully supervised.

A second constraint to this research is the manual correction analysis for the cucumber dataset which was severely limited due to the poor baseline model performance (mAP@50 = 0.038). The pseudo labels generated were too time consuming and difficult to manually correct, making the corrections impractical. Future work should investigate alternative bootstrapping strategies for segmentation tasks, such as starting with larger baseline models or using foundation models like SAM to generate initial pseudo-labels before transitioning to task-specific architectures. This could potentially solve the initial model quality problem that limited the potential of adding ground truth images through manual corrections. In addition to this limitation, this research lacked a detailed analysis of how much faster it is to annotate data from scratch compared to correcting pseudo labels. This comparison could provide more accurate economic projections for industrial deployment and help organizations make better decisions about when to implement pseudo-labeling workflows and when to manually correct pseudo labels to have more ground truth data. Understanding the time difference between these two approaches could bring potential for workflow improvements.

The research was also limited to object detection and instance segmentation tasks with simple model architectures. This means the findings may not generalize to all computer vision applications. Future work should conduct a deeper analysis on the effectiveness of pseudo-labeling on different computer vision tasks such as line detection or semantic segmentation. Additionally, investigating the integration of larger foundation models like YOLO-World or SAM at the initialization phase could potentially improve starting model performance and reduce the ground truth requirements.

The current flow used a full model retraining technique after each expansion of the dataset. One aspect that can be improved is looking into hyperparameter tuning and optimization after expanding the training set instead of retraining from scratch. Future research should investigate whether fine-tuning approaches could improve efficiency compared to full retraining while still maintaining the benefits of using pseudo-labels. The framework should also be tested with modern transformer-based architectures to understand how pseudo-labeling dynamics change with more sophisticated models. The interaction between pseudo-labeling and self-attention mechanisms could reveal new insights into optimal supervision strategies for contemporary computer vision applications.

7.4 Personal Contribution to VBTI

Outside of analyzing the agricultural datasets provided by VBTI and conducting experiments to validate the effectiveness of pseudo-labeling, I have also developed a production-ready pseudo-labeling framework designed specifically for VBTI's OneDL platform that is a contribution that bridges the gap between pseudo-labeling research and practical industrial deployment. The pipeline is a complete and automated system that works seamlessly with VBTI's existing OneDL infrastructure. The program handles everything from data management to model deployment automatically while incorporating a database logging system using SQLite that tracks all iterations, model UIDs, evaluation metrics, and dataset versions across multiple experimental flows. My pipeline automatically manages dataset storage and ensures that pseudo-labeled data, manual corrections, and training datasets are properly saved and versioned within OneDL's cloud-based system, while also including sophisticated CVAT integration that I developed for manual annotation workflows.

I designed the system to handle complex dataset merging operations between ground truth and pseudo-labeled data and built it to support both object detection and instance segmentation tasks with fully parameterized training configurations that work with multiple model architectures including Faster R-CNN, Mask R-CNN, and UPerNet. The pipeline has already added practical value within VBTI's day-to-day operations, as colleagues started approaching me after I finished this research to help them pseudo-label datasets for their own projects. One specific case involved a team member who had developed a strong baseline model that was actually producing pseudo-labels that were higher in quality compared to the ground truth instances. This use case not only helped the research findings but also demonstrated that the pipeline was ready for actual production deployment across VBTI's client projects within the agricultural and manufacturing fields.

VBTI has also recognized the value of the pseudo-labeling framework I developed and is considering integrating it into the front-end of their OneDL platform, which would make the pseudo-labeling workflow right alongside VBTI's existing data management, model training, and evaluation tools to create a more evolved AI ecosystem for their clients. The planned front-end integration would give VBTI's clients and all team members access to the pseudo-labeling techniques I developed and will allow them to implement the semi-supervised learning strategies without requiring deep technical expertise. This aligns with the mission to make AI capabilities accessible to industrial clients who need rapid model development and deployment. The success of my implementation reflects how academic research can translate into tools to help with production to address real industrial challenges.

Chapter 8

Conclusion

The research, experiments, and results within this paper successfully demonstrate the effectiveness of a streamlined pseudo-labeling pipeline with regards to cost optimization and model performance across niche agricultural datasets. By conducting detailed experiments on two real-world use cases involving object detection and instance segmentation, this research was able to establish that pseudo-labeling can achieve substantial cost reductions while maintaining competitive performance in agricultural AI deployment scenarios. The findings reveal that pseudo-labeling workflows not only match fully supervised approaches but can actually outperform them in certain scenarios by allowing models to generalize better on unseen data through the beneficial noise introduced during training. Most importantly, these results provide a clear pathway for organizations to reduce their dependence on expensive external annotation services and help in accelerating model development. The success of this approach in both object detection tasks and complex instance segmentation demonstrates its broad applicability and potential within the agricultural computer vision domain.

Beyond the academic contributions, this research has also produced a practical pseudo-labeling framework that is directly connected to VBTI's OneDL environment. The framework is intended to be utilized within client projects to reduce the cost of outsourcing, bootstrap project development, and build better models within proof-of-concept scenarios where both budget and time constraints are critical factors. The fact that the company is considering integrating the framework within its front-end product further validates how valuable this approach can be and its potential to transform how agricultural AI projects are developed at VBTI and across the AI industry as a whole.

Bibliography

[1] Sama, "Data annotation benefits and advantages in 2023: Sama," January 30 2024.

[2] S. Joshi, "Data annotation: Why it is important for ai/ml success," January 29 2025.

[3] T. AI, "What is data annotation in machine learning?," August 18 2023.

[4] S. Linguae, "The impact of accurate data labeling on model performance," February 2 2024.

[5] G. Karatas, "Data annotation in 2025: Services, types & best practices," October 24 2024.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[7] C. Yun, "Outsourcing data annotation: Challenges & resolutions."

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014.

[9] S. Parker, "Ai proof of concept: Benefits, stages, challenges, and more," February 10 2025.

[10] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou, "Deep learning scaling is predictable, empirically," *arXiv preprint*, 2017.

[11] V. Kapoor, "Looking out for the human in ai & data annotation," September 9 2024.

[12] A. Parti, "Annotation fatigue: Why human data quality declines over time," February 6 2025.

[13] L. Budach, M. Feuerpfeil, N. Ihde, A. Nathansen, N. S. Noack, H. Patzlaff, F. Naumann, and H. Harmouch, "The effects of data quality on machine learning performance," *arXiv preprint*, 2022.

[14] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, 2013.

[15] S. Hu, C.-H. Liu, J. Dutta, M.-C. Chang, S. Lyu, and N. Ramakrishnan, "Pseudoprop: Robust pseudo-label generation for semi-supervised object detection in autonomous driving systems," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4390–4398, 2022.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results." http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[17] Y. Zou, Z. Zhang, H. Zhang, C.-L. Li, X. Bian, J.-B. Huang, and T. Pfister, "Pseudoseg: Designing pseudo labels for semantic segmentation," in *International Conference on Learning Representations (ICLR)*, 2021.

[18] G. Menezes, G. Mazon, R. Ferreira, and V. E. Cabrera, "Artificial intelligence for livestock: a narrative review of the applications of computer vision systems and large language models for animal farming," *Animal Frontiers*, vol. 14, no. 6, 2024.

[19] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language models with self-generated instructions," *arXiv preprint arXiv:2212.10560*, 2022.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015.

[21] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10428–10436, 2020.

[22] J. Choi, S. Kim, N. Kim, S. Oh, and B. Choi, "Kaist multi-spectral day/night data set for autonomous and assisted driving," in *CVPR Workshops*, 2019.

[23] Y. Tian, Y. He, W. Liu, J. Yang, and J. Yan, "Detecting objects in high resolution images by enforcing local consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9547–9556, 2018.

[24] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10687–10698, 2020.

[25] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020.

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Chapter A

Appendix

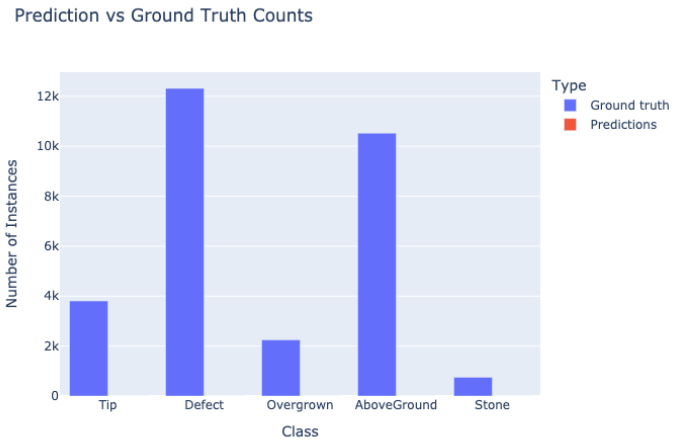


Figure A.1: Instance Distribution for Use Case 1

Table A.1: Use Case 1 - Complete Performance Results Across All Flows and Iterations

Flow	Iter.	GT	PL	mAP@50	mAP@75	mAP(All)	Prec.	Recall	F1	Acc.
F0	0	50	0	0.177	0.057	0.086	0.346	0.771	0.477	0.314
F0	1	200	0	0.281	0.235	0.199	0.325	0.826	0.466	0.304
F0	2	350	0	0.485	0.406	0.363	0.347	0.821	0.488	0.329
F0	3	500	0	0.530	0.490	0.394	0.370	0.825	0.511	0.347
F0	4	650	0	0.431	0.407	0.334	0.378	0.870	0.527	0.359
F0	5	800	0	0.484	0.462	0.374	0.402	0.842	0.544	0.383
F1	5	650	150	0.428	0.410	0.333	0.432	0.849	0.573	0.412
F2	4	500	150	0.451	0.384	0.336	0.420	0.840	0.569	0.396
F2	5	500	300	0.372	0.324	0.276	0.485	0.851	0.618	0.485
F3	3	350	150	0.326	0.283	0.241	0.395	0.820	0.533	0.370
F3	4	350	300	0.394	0.353	0.294	0.403	0.818	0.539	0.375
F3	5	350	450	0.418	0.370	0.309	0.473	0.832	0.603	0.442
F4	2	200	150	0.318	0.277	0.231	0.377	0.830	0.519	0.355
F4	3	200	300	0.279	0.250	0.205	0.460	0.825	0.591	0.426
F4	4	200	450	0.356	0.294	0.248	0.473	0.821	0.597	0.439
F4	5	200	600	0.308	0.289	0.223	0.527	0.829	0.644	0.484
F5	1	50	150	0.222	0.153	0.134	0.484	0.781	0.598	0.443
F5	2	50	300	0.267	0.207	0.163	0.492	0.727	0.587	0.440
F5	3	50	450	0.144	0.097	0.085	0.522	0.696	0.597	0.471
F5	4	50	600	0.115	0.053	0.065	0.474	0.694	0.564	0.427
F5	5	50	750	0.144	0.042	0.073	0.525	0.689	0.596	0.465

GT = Ground Truth Images, PL = Pseudo-Labeled Images

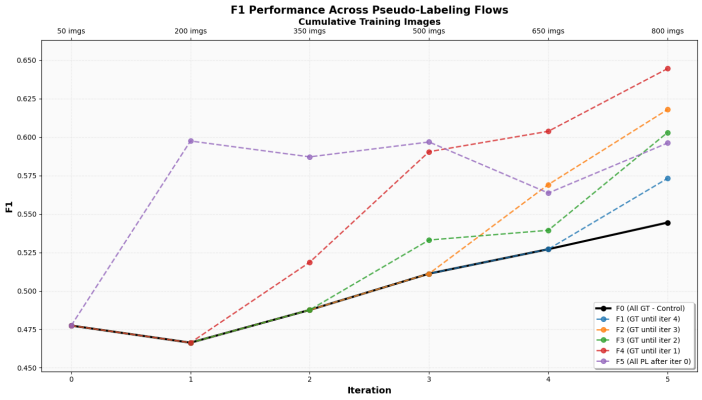


Figure A.2: F1 Distribution for all Flow Iterations in Use Case 1

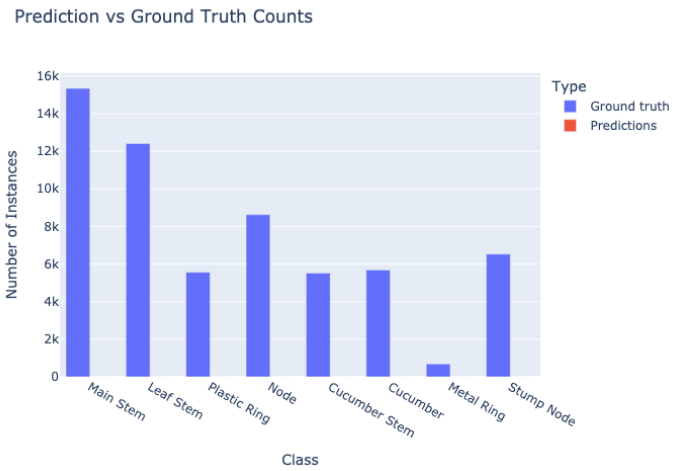


Figure A.3: Class Distribution for Use Case 2

Figure A.4: Instance Distribution for Use Case 2

Table A.2: Use Case 2 - Complete Performance Results Across All Flows and Iterations

Flow	Iter.	GT	PL	mAP@50	mAP@75	mAP(All)	Prec.	Recall	F1	Acc.
F0-S	0	50	0	0.038	0.002	0.012	0.174	0.010	0.020	0.010
F0-S	1	200	0	0.249	0.078	0.108	0.294	0.295	0.295	0.173
F0-S	2	350	0	0.316	0.135	0.155	0.380	0.346	0.362	0.221
F0-S	3	500	0	0.344	0.163	0.177	0.428	0.378	0.402	0.251
F0-S	4	650	0	0.375	0.181	0.196	0.465	0.388	0.423	0.268
F0-S	5	800	0	0.400	0.200	0.210	0.466	0.396	0.428	0.273
F1-S	5	650	150	0.410	0.196	0.214	0.520	0.390	0.446	0.287
F2-S	4	500	150	0.370	0.186	0.195	0.517	0.370	0.431	0.275
F2-S	5	500	300	0.398	0.200	0.211	0.587	0.362	0.448	0.288
F3-S	3	350	150	0.334	0.151	0.170	0.491	0.337	0.400	0.250
F3-S	4	350	300	0.353	0.164	0.181	0.569	0.333	0.420	0.266
F3-S	5	350	450	0.376	0.176	0.194	0.607	0.333	0.430	0.274
F4-S	2	200	150	0.294	0.126	0.142	0.465	0.299	0.364	0.222
F4-S	3	200	300	0.323	0.140	0.160	0.553	0.300	0.389	0.241
F4-S	4	200	450	0.350	0.146	0.172	0.617	0.292	0.397	0.247
F4-S	5	200	600	0.402	0.189	0.208	0.633	0.325	0.429	0.273
F5-S	1	50	150	0.197	0.048	0.080	0.260	0.260	0.260	0.149
F5-S	2	50	300	0.246	0.095	0.117	0.596	0.208	0.308	0.182
F5-S	3	50	450	0.285	0.107	0.133	0.653	0.198	0.304	0.179
F5-S	4	50	600	0.299	0.104	0.135	0.651	0.190	0.294	0.173
F5-S	5	50	750	0.312	0.089	0.133	0.639	0.190	0.293	0.172

GT = Ground Truth Images, PL = Pseudo-Labeled Images

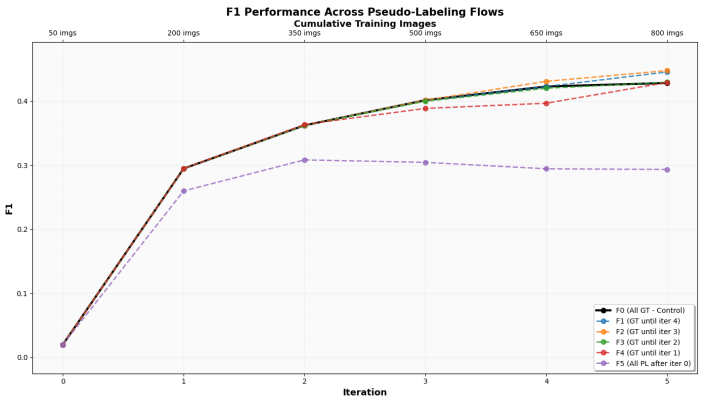


Figure A.5: F1 Distribution for all Flow Iterations in Use Case 2