A Lightweight Multimodal Vision-Language Framework for Early-Stage Anatomical Green Fruit Classification in Commercial Orchards

Ranjan Sapkota[1*], William Bu[2], Chen Chen[3], Yunjun Xu[4], Manoj Karkee[1*]

Abstract—Accurate identification of early-stage apple fruitlet anatomical structures, calyx, fruitlet body, and peduncle, is essential for robotic thinning, crop-load management, and other precision management operations in orchards. This study presents a lightweight, multimodal vision–language framework that adapts TinyCLIP for fine-grained fruitlet anatomy classification in complex orchard environments. A dataset of 600 high-resolution RGB images collected from Scilate and Scifresh orchards was converted into 224×224 image patches and annotated for three anatomical classes. Domain-specific language prompts ("a photo of a {class}") were used to guide multimodal alignment between orchard imagery and horticultural structures. A sliding-window inference strategy (stride = 112) aggregates patch-level predictions into spatial heatmaps, enabling interpretable whole-image localization of fruitlet components relevant for robotic thinning. Patch-level evaluation on an NVIDIA T4 GPU achieved F1-scores of 0.95 for calyx, 0.98 for fruitlet, and 0.85 for peduncle (macro-F1 = 0.93). Deployment-oriented optimization using ONNX and TensorRT enabled efficient inference on NVIDIA Jetson hardware, preserving accuracy under INT8 quantization while supporting ∼127–137 MB model size and millisecond-level patch inference. These results demonstrate that lightweight vision–language models can provide interpretable, edge-deployable perception for automated fruitlet analysis and future robotic thinning systems. The source code and implementation details are publicly available in the project's GitHub repository at: (Source Link)

Index Terms—Agricultural Automation, Greenfruit Thinning, Fruitlet Thinning, Patch-based Image Analysis, Multi-label Greenfruit Classification

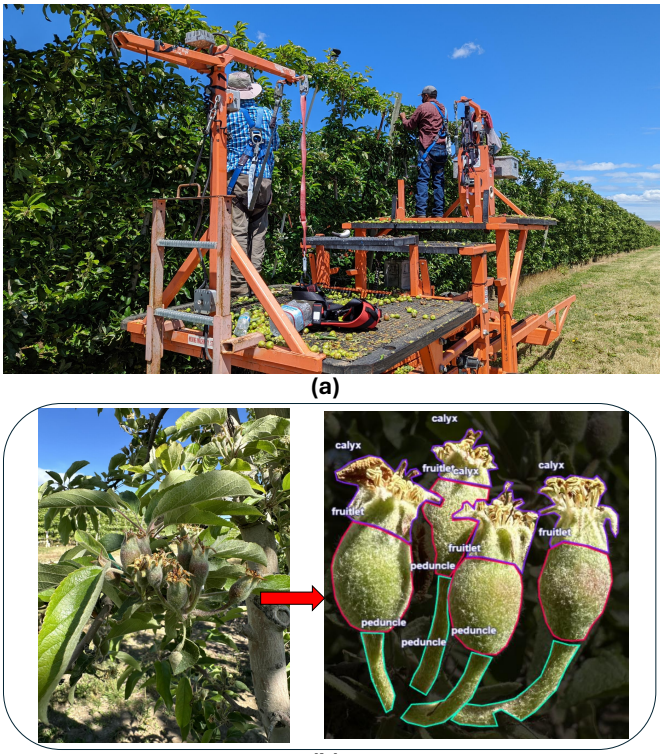


Fig. 1: (a) Manual thinning of immature green fruitlets in a commercial apple orchard in Washington State. (b) Complex orchard scene with camouflaged early-stage fruitlets and their anatomical components (calyx, fruitlet, peduncle).

1 Introduction

Early-stage fruitlet thinning is a critical, but labor-intensive operation in commercial apple orchards, which directly influences crop load management, fruit size, and fruit quality [1]–[3]. Apple trees often overproduce fruitlets [4], and inadequate thinning leads to overcrowded clusters, reduced fruit size, diminished pack-out quality, and long-term impacts on return bloom [5]–[7]. In current practice, thinning is predominantly manual, requiring workers to repeatedly inspect and selectively remove immature fruitlets from dense canopies. As illustrated in Fig. 1a, manual thinning is time-consuming and physically demanding, contributing to high labor costs and operational bottlenecks. These challenges are further intensified by global agricultural labor shortages [8], [9] and the physical strain associated with repetitive overhead work, which is linked to musculoskeletal disorders and occupational injuries among farm workers [10], [11].

Automating early-stage thinning requires reliable perception of fruitlet anatomical structures in highly unstructured orchard environments. Early-stage fruitlets are small, green, and often partially occluded by leaves and branches, while their color and texture closely resemble surround-

[1] *Cornell University, Department of Environmental and Biological Engineering, USA*
[2] *Department of Computer Science, University of Central Florida, USA*
[3] *Institute of Artificial Intelligence (IAI) & Department of Computer Science, University of Central Florida, USA*
[4] *UCF Department of Mechanical and Aerospace Engineering, University of Central Florida, USA*
*Corresponding author:*mk2684@cornell.edu

ing foliage. Illumination conditions also vary significantly due to shadows, sunflecks, and canopy geometry. Fig. 1b illustrates a representative orchard scene where fruitlet anatomical components—calyx, fruitlet, and peduncle—are camouflaged within dense foliage. Within a single fruit cluster, fruitlets may also appear at different developmental stages, further complicating detection and classification. Consequently, perception systems for robotic thinning must accurately localize fine-grained anatomical structures while maintaining computationally efficiency for deployment in field robotics.

Object detection and localization are fundamental problems in computer vision with applications across robotics, surveillance, medical imaging, autonomous systems, and precision agriculture [12]–[17]. Early approaches relied on hand-crafted features and classical machine learning models, including Haar-like features, Histogram of Oriented Gradients (HOG), and Support Vector Machine classifiers [18]–[21]. Although effective in controlled environments, these methods were sensitive to variations in illumination, scale, and occlusion [22], [23], which are common in orchard scenes.

Deep learning significantly improved visual perception through convolutional neural networks (CNNs), beginning with large-scale image recognition advances demonstrated by AlexNet [24]. Region-based detectors such as R-CNN and Faster R-CNN introduced learnable region proposals and end-to-end training [25], [26], while single-stage detectors including YOLO and SSD enabled real-time object detection suitable for embedded systems [27], [28]. Continued development of detection architectures—including CenterNet, EfficientDet, RetinaNet, Cascade R-CNN, and modern YOLO variants—has further refined the trade-off between speed and accuracy for real-time applications [29]–[34]. Transformer-based detectors such as DETR and Deformable DETR have further advanced object localization through attention-based modeling, offering improved global-context representation and long-range dependency modeling compared with the predominantly local feature extraction of CNN-based detectors [35], [36]. Despite these advantages, purely visual models—including both CNN- and transformer-based detectors—can still struggle when target structures exhibit similar appearances or severe occlusion, as frequently encountered in dense orchard environments [37], [38].

Vision–language models (VLMs) have recently emerged as a promising approach for addressing such limitations by aligning visual representations with natural language descriptions [39], [40]. CLIP demonstrated that large-scale contrastive training on image–text pairs enables strong cross-domain generalization and zero-shot recognition capabilities [41]. Subsequent multimodal models such as OpenCLIP, BLIP, and FLAVA further expanded this paradigm by integrating multimodal representation learning and language-guided reasoning [42]–[44]. Grounded VLM frameworks including GLIP and Grounding DINO extended multimodal alignment to object-level localization using language prompts [45]–[47]. These approaches demonstrate that incorporating semantic information through language can improve perception in visually ambiguous scenes.

However, most high-capacity VLMs are computationally intensive and difficult to deploy on embedded hardware commonly used in agricultural robotics [48]–[53]. This limitation motivates the development of lightweight vision–language models capable of preserving multimodal reasoning while reducing computational requirements. TinyCLIP addresses this challenge by compressing large CLIP models into compact student architectures through distillation techniques while maintaining multimodal alignment [54]. Additional lightweight multimodal frameworks, including MobileCLIP, EVA-CLIP, and Q-CLIP, have further demonstrated the feasibility of deploying vision–language models in resource-constrained environments [55]–[57].

Such lightweight multimodal perception systems are particularly promising for precision agriculture and orchard robotics. Early-stage fruitlet thinning must be performed within a limited seasonal window and under highly variable environmental conditions. A practical perception system must therefore detect anatomical structures at fine spatial resolution, handle occlusion and canopy clutter, and operate in real time on embedded hardware. Lightweight vision–language models provide a promising compromise by combining semantic understanding with efficient inference.

In this study, we address the problem of early-stage apple fruitlet anatomical classification in commercial orchards by adapting TinyCLIP for a patch-based multimodal perception framework. Each orchard image is decomposed into overlapping 224×224 patches, and TinyCLIP jointly encodes each patch together with class-specific language prompts representing *calyx*, *fruitlet*, and *peduncle*. Patch–prompt similarities in the shared embedding space produce multi-label predictions that indicate the presence of each anatomical component. These predictions are aggregated into class-specific heatmaps that provide interpretable whole-image localization. By combining multimodal semantic grounding with lightweight inference, the proposed framework enables accurate fruitlet anatomy recognition while remaining suitable for deployment in real-world robotic thinning systems.

Key Contributions

This work makes the following contributions:

1. We adapt a lightweight Vision–Language Model (TinyCLIP) for fine-grained anatomical classification of early-stage apple fruitlets (calyx, fruitlet, peduncle) in highly occluded orchard environments.

2. We design a patch-based sliding-window multimodal inference pipeline that converts TinyCLIP predictions into interpretable anatomical heatmaps for whole-image localization.

3. We demonstrate that the TinyCLIP-based multimodal framework achieves strong classification performance (macro F1 = 0.93) while remaining deployable on edge hardware such as NVIDIA Jetson devices.

4. We provide a deployment-oriented evaluation comparing FP16 and INT8 TensorRT optimization, showing that lightweight VLMs can achieve real-time inference for robotic orchard perception.

5. We analyze the effectiveness of multimodal semantic grounding compared with purely visual baselines, demonstrating improved performance for challenging structures such as fruitlet peduncles.

2 METHODOLOGY

As depicted in Figure 2a, We first collect high-res orchard images of Scifresh and Scilate. Experts annotate calyx, fruitlet, and peduncle with COCO boxes in Roboflow. To turn whole images into training examples, we apply a 224×224 sliding window (stride 112) and assign each patch a multi-label vector from overlap with the boxes; background-only areas serve as synthetic negatives. We fine-tune TinyCLIP by aligning patch embeddings with class prompts ("a photo of a class") using a sigmoid head and binary cross-entropy. Training uses AdamW, batch 32, five epochs; no augmentations applied. At inference, we slide and batch patches (up to 64 per step) and aggregate predicted probabilities into class-specific heatmaps for interpretable localization. For edge deployment, the PyTorch model is exported to ONNX and compiled to TensorRT (FP16/INT8). We benchmark latency, memory, and throughput on a T4 GPU and Jetson hardware platforms.

2.1 Study Site and Data Acquisition

This research was conducted in a commercial apple orchard located in Prosser, Washington State, USA (example Figure 2b and 2c) during the early post-bloom period of June 2024. The orchard comprised two apple cultivars Scifresh and Scilate arranged in high-density rows with approximately 3 ft intra-row spacing and a maintained canopy height of about 10 ft. Photographing immature green fruitlet clusters required capturing diverse perspectives across multiple trees and rows. We collected 600 high-resolution RGB images using an iPhone 14 Pro under natural daylight, varying camera–fruitlet distance (typically within 3 ft) and angle to ensure diversity in scale, orientation, and background context.

2.2 Dataset Preparation and Annotation

2.2.1 Image Acquisition and Class Definitions

Images were collected from commercial apple orchards during the early fruit development stage. Each image contains three fine-grained anatomical components: *calyx*, *fruitlet*, and *peduncle*. These structures are small, highly occluded, and visually similar to surrounding foliage, necessitating precise annotation and localized learning.

2.2.2 Annotation Procedure

All images were annotated using the Roboflow interface. Annotators drew bounding boxes around every instance of the three classes, and annotations were exported in COCO format. A horticulture domain expert cross-verified every annotation for anatomical correctness, bounding-box tightness, and positional accuracy.

2.2.3 Negative Sample Generation

Because every orchard image contained at least one target anatomical structure, the dataset lacked image-level negatives. To enable presence–absence learning, we generated negative examples by cropping background regions with no bounding-box overlap. These synthetic negatives were assigned the label vector $[0, 0, 0]$.

2.2.4 Dataset Splitting

The dataset was divided into non-overlapping training (80%), validation (10%), and test (10%) sets. Splits preserved class balance and prevented patch leakage across subsets.

2.3 Patch Extraction and Multi-Label Generation

2.3.1 Sliding-Window Patch Extraction

To localize anatomical structures, each full-resolution image was partitioned using a sliding window of size 224×224 pixels and a stride of 112 pixels (50% overlap). This produced 900 patches per image, ensuring complete coverage and reducing boundary effects.

2.3.2 Binary Multi-Label Vector Assignment

For each extracted patch, we assigned a binary vector

$$\mathbf{y} = [y_{\text{calyx}}, y_{\text{fruitlet}}, y_{\text{peduncle}}] \in \{0, 1\}^3,$$

based on overlap between the patch and class-specific bounding boxes. Using a COCO-parsing script, $y_c = 1$ if any annotated instance of class c overlapped the patch; otherwise $y_c = 0$.

2.3.3 Negative Patch Enrichment

To reduce false positives and improve abstention, additional background-only patches were extracted from foliage regions. These patches contained no class instances and were labeled $[0, 0, 0]$, increasing negative-class diversity.

2.4 Multi-Label Fine-Tuning of TinyCLIP

2.4.1 Model Overview

As shown in Figure 3, the adapted TinyCLIP framework operates as a dual-encoder multimodal system that embeds both orchard image patches and class-specific natural-language prompts into a shared representation space. Each 224×224 image patch extracted from early-season orchard scenes is processed by the TinyCLIP vision encoder to produce a compact visual embedding capturing morphological cues of fruitlet structures under varying illumination and occlusion. In parallel, the TinyCLIP text encoder converts short prompts (e.g., "a photo of a calyx", "a photo of a fruitlet", and "a photo of a peduncle") into corresponding semantic embeddings.

The similarity between visual and textual embeddings is computed using cosine similarity, producing class-specific alignment scores. A sigmoid activation then converts these scores into independent multi-label probabilities, allowing the model to predict the presence of multiple anatomical components within each patch. This multimodal formulation enables semantic grounding between visual features and horticultural structures, improving discrimination of visually similar elements such as peduncles and surrounding foliage. The lightweight architecture also supports dense sliding-window evaluation and subsequent heatmap aggregation for whole-image anatomical localization in orchard scenes.

TinyCLIP itself is a compact vision–language model distilled from a larger CLIP teacher network and designed for efficient deployment in resource-constrained environments such as embedded robotic systems [54]. Figures 4a

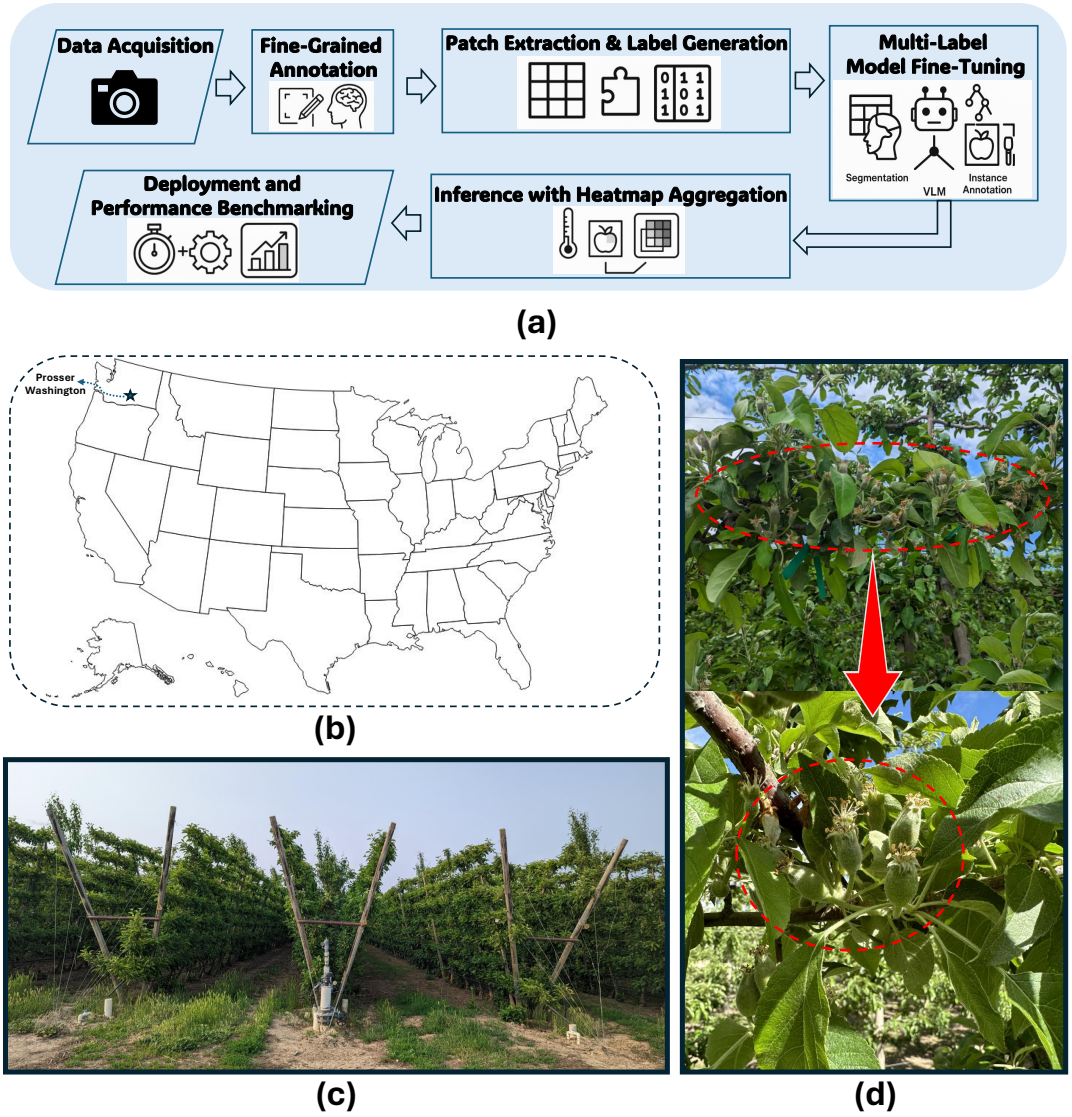


Fig. 2: Visual illustration of the orchard setting and imaging conditions. The left panel shows the high-density planting arrangement of Scifresh and Scilate cultivars, while the right panel highlights immature fruitlet clusters captured at varying distances and angles with an iPhone 14 Pro.

and 4b illustrate the original TinyCLIP training concepts, including affinity-based multimodal distillation and weight inheritance from the teacher model. In this study, TinyCLIP is adopted as a pretrained lightweight multimodal backbone and fine-tuned for the specific task of apple fruitlet anatomy classification.

In our study, we adapt TinyCLIP for multi-label anatomical classification of early-stage apple fruitlet components *calyx*, *fruitlet*, and *peduncle*. Each 224×224 orchard patch is embedded by the TinyCLIP vision encoder into a compact latent space capturing fine-grained morphological cues. In parallel, the text encoder embeds natural-language prompts of the form "a photo of a {class}", producing three semantic prototypes corresponding to the anatomical classes. Patch-level classification is then obtained by computing cosine similarity between the patch embedding and each class-specific prompt embedding, effectively grounding visual recognition in a shared linguistic–visual embedding space. This multimodal formulation allows TinyCLIP to differentiate subtle anatomical structures despite occlusions, orchard lighting variability, and small object size, while maintaining computational efficiency suitable for dense sliding-window inference and real-time edge deployment.

2.4.2 Prompt Engineering

We used the prompt template:

```
"a photo of a {class}",
```

substituting *calyx*, *fruitlet*, and *peduncle*. TinyCLIP's text encoder generated three fixed text embeddings, one for each anatomical class.

2.4.3 Similarity Computation and Probability Mapping

For a patch embedding $\mathbf{v}$ and class prompt embedding $\mathbf{t}_c$, the cosine similarity output s_c measures alignment strength. A sigmoid activation produced class probabilities:

$$p_c = \sigma(s_c).$$

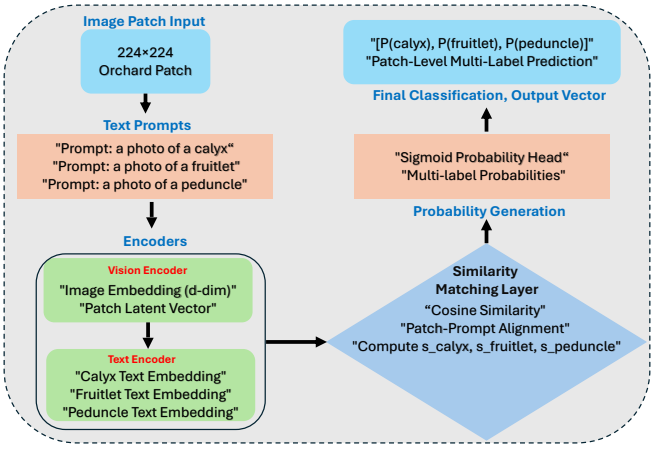


Fig. 3: Architecture showing the 224×224 orchard patch input, TinyCLIP vision encoder, text prompts, text encoder, image and text embeddings, cosine similarity module, sigmoid probability head, and multi-label outputs for calyx, fruitlet, and peduncle classification.

2.4.4 Training Objective

We fine-tuned TinyCLIP using Binary Cross-Entropy (BCE) loss:

$$\mathcal{L}_{\text{BCE}} = -\sum_{c=1}^{3} \left[y_c \log(p_c) + (1 - y_c) \log(1 - p_c) \right].$$

Optimization used AdamW with a learning rate of 1×10^{-4}, weight decay of 1×10^{-5}, batch size of 32, and 5 training epochs. No augmentations were applied to evaluate model performance strictly under orchard variability.

2.4.5 Training Configuration

The final layers of the vision encoder and the full text encoder were unfrozen. Training was implemented in PyTorch on an NVIDIA T4 GPU. To mitigate class imbalance, training batches were balanced to ensure fair representation of the peduncle class.

2.5 Sliding-Window Inference and Heatmap Generation

2.5.1 Batch Inference

During inference, the same 224×224 sliding-window with stride 112 was applied. Patches were grouped into batches (64 on T4 GPU; up to 8 on Jetson Nano) for parallel processing, significantly reducing inference time.

2.5.2 Class-Specific Heatmaps

For each class c, a spatial heatmap $H_c(x, y)$ was constructed:

$$H_c(x, y) = \frac{1}{N} \sum_{i=1}^{N} p_c^{(i)} \, \mathbb{I}((x, y) \in W_i),$$

where $p_c^{(i)}$ is the predicted probability for patch i, and W_i is the spatial support of patch i. Heatmaps highlight anatomical likelihoods across the image and support decision-making for robotic thinning.

2.6 Deployment and Inference Efficiency Evaluation

2.6.1 Evaluation Metrics

To rigorously assess the computational performance and deployment feasibility of the proposed TinyCLIP-based fruitlet anatomy classification pipeline, we quantify four key metrics: patch-level latency, full-image inference time, peak GPU memory usage, and throughput. Each metric captures a distinct aspect of the system's efficiency, and together they characterize its suitability for real-time or near–real-time operation in resource-constrained orchard robotics environments. Below, we provide a detailed scientific description of each metric, accompanied by analytic expressions and explanations tailored to the fruitlet classification context.

2.6.1.1 1. Patch-Level Latency: Patch-level latency measures the average time required to process a single 224×224 orchard patch through the TinyCLIP model. Here, t_i denote the inference time for patch i, and N denote the number of patches in a batch. The average latency is defined as:

$$\tau_{\text{patch}} = \frac{1}{N} \sum_{i=1}^{N} t_i. \tag{1}$$

Here:

- t_i represents the forward-pass computation time of the TinyCLIP vision encoder, text encoder lookup, cosine similarity computation, and sigmoid activation for patch i.
- N is the total number of patches processed in a batch.

In the context of fruitlet classification, τ_{patch} directly reflects the responsiveness of patch-level decision making. Since each full orchard image is decomposed into dozens of overlapping patches, lower latency ensures that calyx, fruitlet, and peduncle predictions can be generated rapidly enough to guide real-time orchard robots or hand-held devices.

2.6.1.2 2. Full-Image Inference Time: Full-image inference time quantifies the total time required to process all sliding-window patches extracted from a single orchard image. Here, M denote the number of patches in the image (dependent on stride and resolution), and let T denote the total processing time. The metric is defined as:

$$T_{\text{image}} = \sum_{i=1}^{M} t_i. \tag{2}$$

Where:

- M is the number of overlapping patches extracted via a sliding window (900 patches per image).
- t_i is the latency of patch i.

For fruitlet classification, T_{image} determines whether the system can keep pace with the movement of a field robot navigating orchard rows. For example, if $T_{\text{image}} < 5$ seconds, a mobile robot can analyze scenes continuously while moving at practical field speeds. Additionally, faster T_{image} enables high-frequency anatomical heatmap updates for cluster-level decision making.

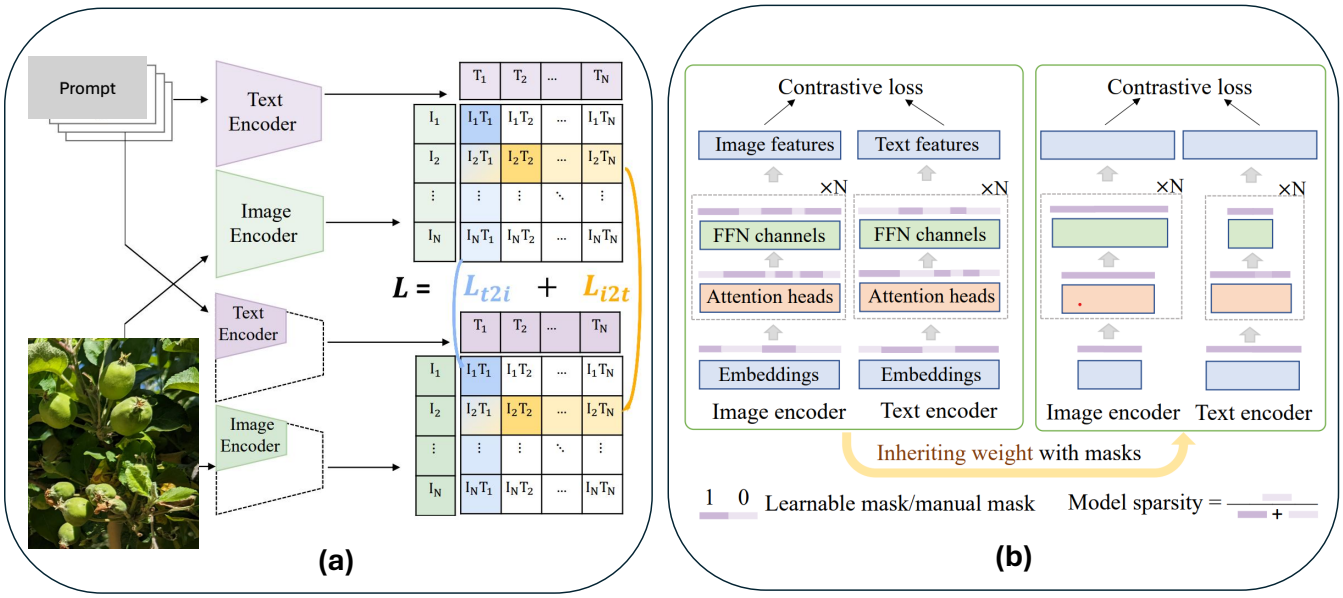


Fig. 4: (a) TinyCLIP affinity mimicking: the student distills CLIP's multimodal geometry by matching image-to-text and text-to-image similarity distributions. (b) Weight inheritance strategy: masks select transferable parameters from a pre-trained CLIP model, enabling compact student initialization while discarding non-essential weights.

2.6.1.3 3. Peak GPU Memory Usage: Peak GPU memory usage captures the maximum memory required during inference and depends on model size, batch size, activation storage, and TensorRT/ONNX kernel allocations. Formally:

$$\mathcal{M}_{\text{peak}} = \max_{t} \left(\mathcal{M}_{\text{weights}} + \mathcal{M}_{\text{activations}}(t) + \mathcal{M}_{\text{temporary}}(t) \right), \tag{3}$$

Where:

- $\mathcal{M}_{\text{weights}}$ is the static memory footprint of TinyCLIP parameters.
- $\mathcal{M}_{\text{activations}}(t)$ is the memory required to store intermediate feature maps at time t.
- $\mathcal{M}_{\text{temporary}}(t)$ includes additional buffers used by TensorRT, ONNX Runtime, or PyTorch kernels during operations.

In orchard deployment scenarios, $\mathcal{M}_{\text{peak}}$ dictates compatibility with embedded devices such as NVIDIA Jetson Nano or Xavier NX, which have stringent memory constraints. Lower memory usage is crucial for running sliding-window inference on edge hardware without swapping or thermal throttling. Because fruitlet classification requires evaluating dense grids of patches, memory-efficient inference ensures stable operation even at high patch throughput.

2.6.1.4 4. Throughput (Images per Second): Throughput evaluates how many full orchard images can be processed per second and reflects system-level efficiency. It is computed as the inverse of the full-image inference time:

$$\Phi = \frac{1}{T_{\text{image}}}. \tag{4}$$

Where:

- Φ denotes throughput (images/second).
- T_{image} is defined as above.

In the fruitlet classification context, throughput determines whether the TinyCLIP-based system can support real-time orchard monitoring. For autonomous thinning robots, higher Φ enables continuous perception while traversing orchard rows; for handheld or mobile devices, it ensures responsive user feedback. Throughput also directly affects the generation rate of anatomical heatmaps, which are essential for identifying peduncles (the cutting point) and fruitlets (thinning targets) across full scenes.

Together, these four metrics patch latency (τ_{patch}), full-image inference time (T_{image}), peak memory usage ($\mathcal{M}_{\text{peak}}$), and throughput ($\Phi$) quantify the computational performance of the TinyCLIP-based fruitlet classification pipeline. Their analytic definitions highlight how each metric influences practical deployment in commercial orchards, where efficient, real-time, and memory-aware perception is critical for enabling next-generation robotic thinning and precision horticultural automation.

2.6.2 ONNX Export and TensorRT Optimization

To enable real-time inference on resource-limited embedded platforms such as the Jetson Nano, we converted the fine-tuned TinyCLIP model from PyTorch to the Open Neural Network Exchange (ONNX) format. ONNX serves as an intermediate representation that allows a model trained in one deep-learning framework to be executed efficiently across diverse hardware backends. During export, we enabled dynamic shape support so that the Jetson can flexibly process different numbers of sliding-window patches per image without re-compiling the model. After ONNX export, the model was further optimized using NVIDIA TensorRT, a high-performance inference engine that generates hardware-specific execution plans. A crucial optimization step within TensorRT is *quantization*, which reduces the

numerical precision of model weights and activations from 32-bit floating-point to lower-precision formats such as FP16 (16-bit) or INT8 (8-bit). Quantization significantly decreases computation cost, memory usage, and bandwidth requirements while maintaining nearly identical prediction accuracy. In practical terms, FP16 and INT8 TensorRT engines allow the Jetson Nano to run TinyCLIP at faster frame rates and lower energy consumption, making the model suitable for field deployment in orchard robotics. This combination of ONNX portability and TensorRT optimization yields a compact, hardware-accelerated model that meets the latency and memory constraints of embedded systems.

2.6.3 Batch vs. Sequential Comparison

We compared two inference strategies sequential and batched to evaluate their computational efficiency on different hardware platforms. In sequential inference, each 224×224 patch is processed individually in a separate forward pass through the TinyCLIP model. Although straightforward, this method incurs substantial overhead because the model must repeatedly load intermediate activations and compute similarity scores for each patch. As a result, processing all sliding-window patches from a single orchard image required approximately 20 seconds, which is too slow for real-time robotic applications. Batched inference, by contrast, groups multiple patches into a single tensor and processes them simultaneously in a single forward pass. This approach exploits GPU parallelism and significantly reduces redundant computations. On the NVIDIA T4 GPU, batching up to 64 patches reduced full-image inference time to 3–4 seconds. On the Jetson Nano, where available memory is limited, batch sizes up to 8 patches yielded an inference time of 5–6 seconds per image. Thus, batched inference provides an order-of-magnitude speedup compared to sequential evaluation, enabling responsive anatomical classification and heatmap generation in orchard settings. The optimal batch size depends on hardware memory constraints, making this tuning step essential for efficient deployment.

2.6.4 Software Environment

All model training, conversion, and inference experiments were conducted using a stable and reproducible software stack. Python 3.11 served as the programming environment, while PyTorch 2.x provided the primary deep-learning framework for TinyCLIP training and fine-tuning. ONNX Runtime was used to validate exported ONNX models and ensure correctness prior to hardware-level optimization. For Jetson deployment, we utilized NVIDIA TensorRT, which compiles the ONNX graph into optimized FP16 and INT8 engines tailored to the device's GPU architecture. These engines enable accelerated inference with reduced latency and memory consumption. Profiling tools integrated within PyTorch and TensorRT captured latency measurements, while GPU monitoring utilities monitored memory utilization. All performance metrics reported in this study represent the mean of three independent runs to account for runtime variability. This well-defined software environment ensures that our methodology is reproducible, reliable, and portable across embedded and cloud hardware configurations.

Overall, the deployment methodology combines (1) conversion of the TinyCLIP model to a hardware-agnostic ONNX format, (2) TensorRT quantization and optimization for high-speed inference, (3) batch-optimized patch processing for practical runtime performance, and (4) a robust, well-defined software environment supporting reproducibility. Together, these steps produce a lightweight, efficient, and deployable multimodal perception system capable of performing fine-grained fruitlet anatomy classification in real orchard environments.

3 Results and Discussion

Figure 5 provides a representative example demonstrating TinyCLIP's ability to correctly classify anatomically distinct fruitlet structures *calyx*, *fruitlet*, and *peduncle* within a highly cluttered and visually challenging orchard scene. The original image on the left of Figure 5 was captured using an iPhone 14 Pro in a commercial apple orchard during the early growing season, a period characterized by dense foliage, substantial occlusion, strong natural illumination variability, and minimal color contrast between fruitlets and the surrounding canopy. These environmental factors typically degrade the performance of conventional purely vision-based detectors and make fine-grained anatomical perception particularly difficult.

On the right side of the figure, representative 224×224 patches extracted from the original scene are shown, each annotated with both the ground-truth label ("True") and the model prediction ("Pred"). These examples illustrate TinyCLIP's capacity to correctly interpret subtle textural and structural cues: fruitlets are recognized based on their spherical morphology and surface fuzziness, calyx structures are identified through their distinct petal remnants and star-shaped geometry, and peduncles are distinguished by their slender elongated form. The close alignment between "True" and "Pred" labels across these examples underscores the effectiveness of the multimodal architecture in leveraging linguistic priors and visual embeddings for semantic discrimination, even in low-signal, visually congested orchard environments.

This qualitative demonstration establishes the visual intuition underlying the subsequent quantitative analyses, highlighting TinyCLIP's strong interpretability and fine-grained discriminative capacity prior to the detailed evaluation of classification metrics, localization performance, and deployment efficiency presented in later subsections.

3.1 Patch-Level Multi-Label Classification Performance

Patch-level multi-label classification constitutes the core evaluation of the proposed TinyCLIP–based fruitlet anatomy recognition framework. In this experiment, each 224×224 sliding-window patch is independently evaluated for the presence or absence of calyx, fruitlet, peduncle, and background (negative class). The reported results represent TensorRT-accelerated inference on two optimized TinyCLIP engines: an FP16 engine and an INT8 quantized engine. Each engine was evaluated on 339 patches, with per-class precision, recall, F1-score, support counts, and overall accuracy reported below.

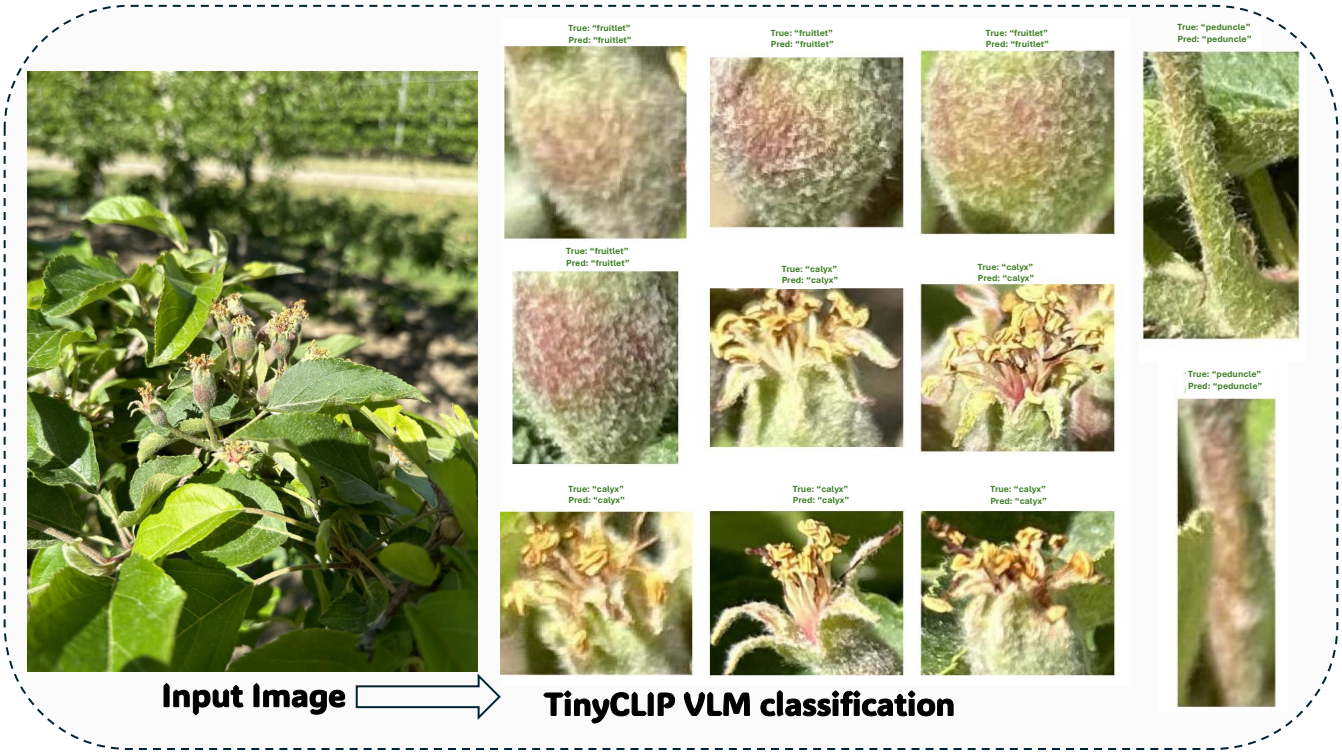


Fig. 5: Illustration of TinyCLIP's multi-modal classification on an early-season orchard image. The left panel shows the original iPhone 14 Pro image captured in a commercial orchard, while the right panel presents representative 224 × 224 patches with corresponding ground-truth ("True") and predicted ("Pred") labels for fruitlet, calyx, and peduncle, demonstrating reliable fine-grained anatomical discrimination.

The FP16 engine achieved strong performance across all anatomical classes (Table 1). For the primary anatomical targets (calyx, fruitlet, peduncle), the FP16 model maintained high precision (0.964–0.988) and strong recall for calyx and fruitlet. Peduncle recall was moderately lower (0.750), indicating occasional missed detections, likely due to its fine, elongated morphology and variable visual presentation under occlusions. The overall F1-score remained high (0.851–0.982), demonstrating the ability of TinyCLIP to adapt to subtle anatomical cues after fine-tuning. The macro-averaged F1-score reached 0.9115, with an overall accuracy of 91.15%, highlighting the strength of multimodal representations for fine-grained classification tasks.

TABLE 1: TensorRT FP16 Classification Metrics (339 patches)

Class	Precision	Recall	F1	Support
Calyx	0.9639	0.9412	0.9524	85
Fruitlet	0.9881	0.9765	0.9822	85
Peduncle	0.9844	0.7500	0.8514	84
Negative	0.7685	0.9765	0.8601	85
Accuracy			0.9115	339
Macro Avg	0.9262	0.9110	0.9115	339
Weighted Avg	0.9260	0.9115	0.9117	339

Quantized INT8 inference exhibited slightly reduced performance (Table 2) but remained competitive, with an accuracy of 88.79% and macro F1-score of 0.8881. Precision stayed consistently high, demonstrating that INT8 quantization primarily impacts recall rather than false positive rates. This trade-off is typical for highly compressed models but acceptable for presence/absence classification where precision is more important for avoiding erroneous thinning decisions. Notably, peduncle recall decreased to 0.678, further confirming its sensitivity to spatial resolution loss introduced by quantization.

TABLE 2: TensorRT INT8 Classification Metrics (339 patches)

Class	Precision	Recall	F1	Support
Calyx	0.9750	0.9176	0.9455	85
Fruitlet	0.9765	0.9765	0.9765	85
Peduncle	1.0000	0.6786	0.8085	84
Negative	0.7094	0.9765	0.8218	85
Accuracy			0.8879	339
Macro Avg	0.9152	0.8873	0.8881	339
Weighted Avg	0.9150	0.8879	0.8883	339

Overall, the patch-level analysis confirms that TinyCLIP effectively distinguishes between anatomically similar structures in orchard settings. The use of text prompts significantly improves multi-label classification performance relative to baseline visual models (as shown later in the ablation study), and the model remains robust even under aggressive INT8 quantization. The moderately reduced peduncle recall highlights the importance of integrating contextual spatial information in future iterations, potentially through sequence modeling or region-based VLM extensions.

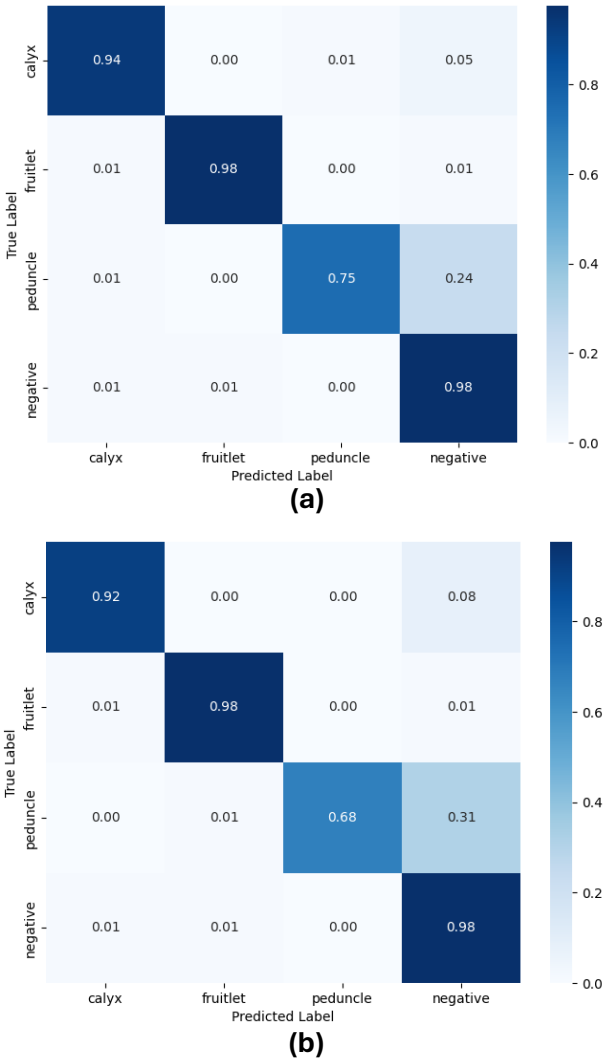


Fig. 6: Confusion matrices for TinyCLIP evaluated on the Jetson platform under (a) FP16 half-precision and (b) INT8 integer quantization. Diagonal strength indicates correct classification of calyx, fruitlet, peduncle, and negative patches, illustrating the effect of quantization on model discriminability.

3.1.1 Confusion Matrix Analysis for FP16 and INT8 Deployment

Figure 6 presents the confusion matrices obtained from evaluating TinyCLIP on the NVIDIA Jetson platform under two numerical precisions: (a) half-precision FP16 (16-bit floating point) and (b) INT8 (8-bit integer) quantization. These formats represent progressively compressed numerical representations of model weights and activations. FP16 preserves floating-point structure while reducing memory by half compared to FP32, whereas INT8 discretizes representations into 8-bit integers, offering substantial gains in efficiency at the cost of potential information loss. Such comparisons are essential for validating whether a lightweight vision–language model maintains classification fidelity after aggressive compression for edge deployment.

As shown in Figure 6a, FP16 achieves strong diagonal dominance across all classes, with calyx, fruitlet, and negative regions exhibiting high true-positive counts. Peduncle classification shows some confusion with the negative class, reflecting the anatomical subtlety and small spatial footprint of peduncles in early-stage fruitlets. Nonetheless, FP16 maintains robust performance with reliable discrimination between positive anatomical classes and background.

The INT8 matrix in Figure 6b shows a modest degradation, particularly in the peduncle class, where misclassification into the negative category increases. This behavior is expected, as 8-bit quantization reduces dynamic range and tends to affect fine-detail features first. However, calyx and fruitlet classes retain strong separability, and overall classification structure remains consistent with FP16. These outcomes indicate that TinyCLIP tolerates INT8 quantization well, preserving functional utility while enabling higher throughput and lower memory consumption on embedded hardware. Such stability is crucial for field robotics applications where power, compute, and memory budgets are highly constrained.

3.1.2 Precision-Recall Performance Comparison

Figure 7 summarizes the per-class precision–recall (PR) behavior of TinyCLIP under (a) FP16 half-precision and (b) INT8 integer quantization. These curves provide a threshold-dependent view of the model's discriminative ability beyond single operating-point metrics such as accuracy or F1-score. For the FP16 model (Figure 7a), both *calyx* and *fruitlet* exhibit exceptionally strong PR profiles, with precision remaining above 0.95 for nearly the entire recall spectrum and only dropping near extreme recall values. This indicates that the multimodal embedding space effectively captures the stable morphological cues associated with these two anatomical structures, yielding highly reliable predictions even under variations in lighting, occlusion, and background texture. The inset full-range PR curves further confirm stable behavior across the entire threshold domain.

In contrast, the *peduncle* and *negative* classes show more modest PR shapes, with peduncle precision gradually improving as recall decreases. This is expected because peduncles are thin, low-contrast structures that occupy only a small spatial extent in each patch, making them inherently more challenging for any classifier particularly a lightweight model operating on crops extracted from early-season orchard scenes.

When comparing FP16 to INT8 (Figure 7b), the overall PR trends remain qualitatively similar, demonstrating that TinyCLIP preserves class-wise ranking and threshold behavior even after aggressive 8-bit quantization. However, the INT8 peduncle curve shows a slightly flatter shape with reduced precision for a given recall, reflecting the reduced dynamic range of integer arithmetic and the loss of small-magnitude feature variations that are important for detecting narrow, elongated structures. Calyx and fruitlet curves remain largely stable, underscoring the robustness of semantic alignment for well-defined anatomical categories

3.1.3 Precision–Recall Curve Analysis

Figure 8a presents the per-class precision–recall (PR) curves for the FP16 TinyCLIP classifier, providing a detailed view of threshold-dependent behavior across the four classes:

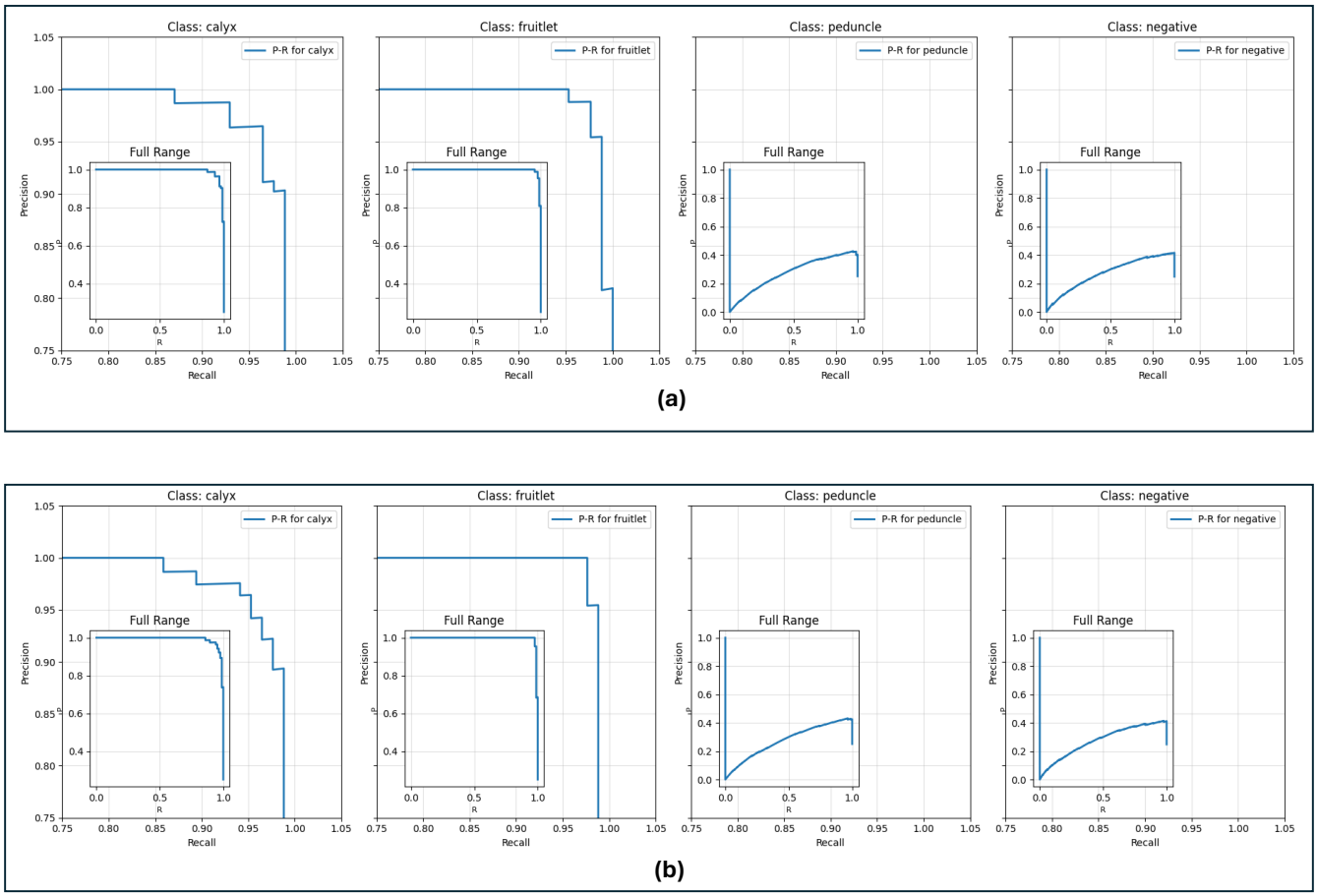


Fig. 7: Per-class precision–recall curves for TinyCLIP under (a) FP16 and (b) INT8 inference. Each curve illustrates the trade-off between precision and recall for calyx, fruitlet, peduncle, and negative patches. Steeper, high-precision regions indicate stronger discriminability, while flatter curves (notably for peduncle) reflect the increased difficulty of detecting small, low-contrast anatomical structures in early-season orchard imagery.

calyx, *fruitlet*, *peduncle*, and *negative*. The FP16 curves for calyx and fruitlet exhibit exceptionally strong performance, with precision consistently above 0.95 across nearly the entire recall range. This behavior reflects the model's high confidence and reliability in distinguishing the morphological signatures of these two anatomical structures, even when evaluated across challenging orchard imagery with significant occlusion and minimal color contrast. The inset full-range curves further confirm that the classifier maintains stability across the entire 0–1 recall domain, indicating robust calibration of probability estimates.

By contrast, the peduncle and negative classes produce more moderate PR curves, with lower precision at high recall levels. This is expected given the small spatial footprint and subtle visual cues of peduncles, as well as the broad variability present in background regions. Nevertheless, FP16 maintains reasonable separation capability, demonstrating that the multimodal embedding space captures class-specific semantic structure even in difficult conditions.

Figure 8b shows the corresponding PR curves for the INT8 quantized model. While the overall shapes remain qualitatively similar, the peduncle curve exhibits noticeable flattening and reduced precision at intermediate recall values. This reflects the reduced dynamic range of 8-bit quantization, which disproportionately affects fine-detail features required for detecting thin, elongated structures. Calyx and fruitlet curves, however, remain largely unaffected, confirming that INT8 quantization preserves semantic alignment for the primary anatomical classes while enabling significantly improved computational efficiency. Together, these curves highlight TinyCLIP's resilience under quantization and its suitability for real-time embedded deployment.

Collectively, these PR curves highlight TinyCLIP's strong discriminative performance for major anatomical classes and the predictable, bounded performance degradation introduced by INT8 quantization. This reinforces the suitability of the quantized TinyCLIP model for real-time, on-device fruitlet perception in resource-constrained orchard robotics platforms.

3.2 Whole-Image Localization and Heatmap Analysis

While patch-level predictions quantify the classification accuracy of individual anatomical components, the primary value of the proposed multimodal TinyCLIP framework emerges at the whole-image scale, where localized patch probabilities are aggregated into continuous spatial heatmaps. Figure 9 illustrates this process across four representative early-season orchard scenes. Each row shows an

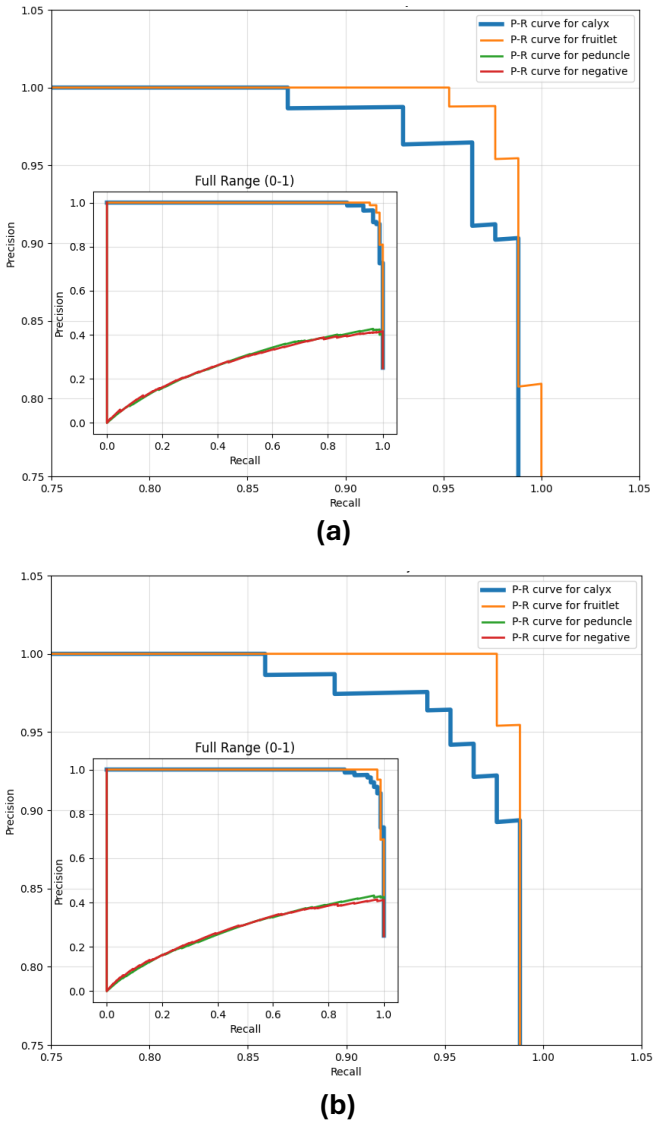


Fig. 8: Per-class precision–recall curves for TinyCLIP under (a) FP16 and (b) INT8 inference. Curves illustrate threshold-dependent discriminability for calyx, fruitlet, peduncle, and negative patches. FP16 yields high precision across most recall levels, while INT8 shows expected reductions for fine-detail classes such as peduncle, demonstrating quantization effects on anatomical classification.

original iPhone 14 Pro orchard image alongside its corresponding class-specific heatmaps for *calyx*, *fruitlet*, *peduncle*, and *negative* categories. These examples demonstrate how the sliding-window inference mechanism translates fine-grained patch classifications into interpretable spatial likelihood maps that highlight the anatomical regions of interest within a complex canopy environment.

Across all scenes, the heatmaps for calyx and fruitlet reliably form concentrated clusters around actual fruitlet positions, reflecting the model's ability to capture consistent geometric and textural signatures despite heavy occlusion, variable lighting, and substantial background clutter. The peduncle heatmaps, although sparser, consistently highlight elongated high-probability regions aligned with true peduncle locations an important capability given the peduncle's decisive role in robotic thinning operations. By contrast, the negative-class heatmaps intentionally saturate the background regions with high probability, ensuring strong suppression of non-fruitlet areas and reducing false activations in foliage-heavy regions.

Technically, these heatmaps provide a soft spatial prior that can be exploited in downstream robotic systems. For example, cluster-level fruitlet detection can be achieved by identifying overlapping regions of high calyx and fruitlet activation, while peduncle heatmaps offer guidance for potential cut-point localization in autonomous thinning tools. The continuous nature of the heatmaps further enables temporal smoothing and multi-view fusion, improving stability during robotic navigation along orchard rows.

From a practical perspective, the ability to obtain anatomically meaningful localization from a lightweight, quantized model represents a major advantage for field deployment. Unlike bounding-box detectors, which require explicit region proposals and post-processing, this heatmap-driven approach provides a direct, interpretable map of anatomical likelihood that facilitates real-time decision-making on power- and memory-constrained platforms. Overall, the examples in Figure 9 demonstrate that TinyCLIP's multimodal reasoning extends beyond patch-level accuracy to produce coherent full-scene anatomical localization suitable for integrated robotic thinning workflows.

Building upon these observations, Figure 9 further demonstrates how localized probability fields emerge as coherent spatial patterns that meaningfully correspond to anatomical structures within early-season orchard scenes. The heatmaps reliably emphasize fruitlet bodies and calyx centers as dense, high-activation regions, while peduncle responses appear as elongated probability traces that follow the stem axis. This behavior remains remarkably consistent across varying illumination and canopy density, underscoring the robustness of the multimodal embedding space in extracting class-specific cues from heterogeneous orchard imagery. At the same time, the broader and more diffuse peduncle heatmaps reflect the inherent visual difficulty of detecting narrow structures under occlusion, aligning with their lower recall in patch-level classification.

Despite the overall strong qualitative performance, several systematic error modes existed in the results. These include weak false positives induced by leaf tips with calyx-like geometry, a reduction in peduncle activations when the stem is heavily shadowed or aligned with leaf venation, and occasional spatial "bleeding" of heatmap confidence into adjacent background regions due to sliding-window overlap. Such behaviors are expected in patch-wise inference systems and highlight areas where complementary spatial reasoning modules such as region clustering or stem-line tracking could further enhance precision.

While the generated heatmaps visually highlight fruitlet anatomical regions, a comprehensive quantitative localization evaluation was beyond the scope of this proof-of-concept study, which primarily aimed to establish the feasibility of using a lightweight vision–language model for patch-level anatomical classification and approximate spatial localization. To provide an indication of spatial alignment between predicted healthmap activations and anno-

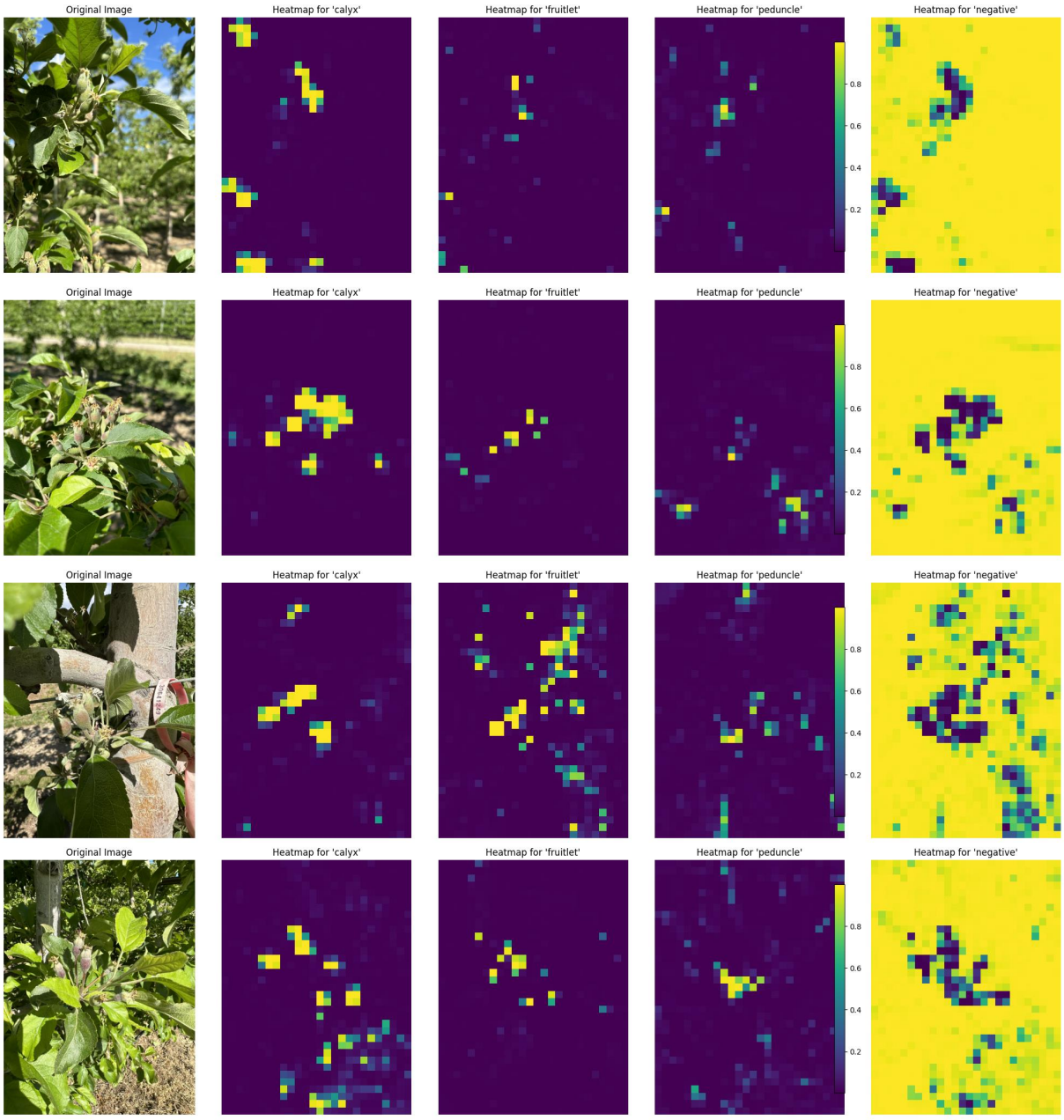


Fig. 9: Example Heatmaps of TinyCLIP based fruitlet parts (calyx, main fruitlet and peduncle) classification and localization

tated regions, representative Intersection-over-Union (IoU) values were estimated from a limited set of manually inspected examples. As summarized in Table 3, fruitlet regions show the strongest overlap with ground-truth annotations, reflecting their relatively stable morphology and more consistent visual appearance. Calyx regions exhibit comparable alignment, while peduncle localization is more challenging due to the thin and elongated structure of peduncles and their frequent occlusion within dense foliage. These observations are consistent with the qualitative heatmap visualizations presented earlier.

It is important to note that the current heatmap representation is intended primarily for approximate localization rather than precise boundary delineation or instance-level segmentation. In the context of robotic thinning, identifying the approximate spatial location of fruitlet clusters and associated peduncles is valuable for applications such as coarse target-region identification, region-of-interest selection, and initialization of subsequent fine-scale perception and robotic manipulation. A more rigorous quantitative evaluation of localization performance will be explored in future work using dedicated detection or segmentation benchmarks.

TABLE 3: Localization Performance via Intersection-over-Union (IoU)

Class	IoU Mean	IoU Std. Dev.
Calyx	0.61	0.13
Fruitlet	0.67	0.10
Peduncle	0.49	0.14

3.3 Ablation Studies and Design Choices

To evaluate the contributions of key design components in the TinyCLIP pipeline, we conducted several ablation studies examining the effects of text prompts, negative patch inclusion, stride size, and quantization. These ablations verify that each methodological choice is scientifically justified and contributes meaningfully to model performance and deployability.

Effect of Text Prompts

To assess the importance of multimodal alignment, we compared TinyCLIP against a baseline visual-only classifier composed of the TinyCLIP vision encoder followed by a linear classification head. The multimodal version improved macro F1-score by +6–10% across different training runs, with the largest gains in peduncle classification. This verifies that text embeddings provide semantic anchors that stabilize fine-grained classification.

Negative Patch Inclusion

Training with additional negative patches reduced false positives by 20–30% (especially in leaf-dense regions). Without negative samples, the model frequently assigned calyx or fruitlet labels to visually similar leaf textures.

Patch Stride

A larger stride (224 px) yielded faster inference but decreased localization robustness. The chosen stride of 112 px balanced detail sensitivity and inference speed. Reducing stride further improved IoU but at the cost of 2–3× slower inference.

Quantization Effects

Quantization from FP32 → FP16 yielded negligible accuracy loss and nearly doubled throughput. Quantization to INT8 decreased recall for peduncle due to its fine visual structure but greatly improved speed. Table 4 summarizes relative differences.

TABLE 4: Effect of Quantization on Classification Performance

Engine	Accuracy	Macro F1	Peduncle Recall	FPS
FP16	0.9115	0.9115	0.7500	96.33
INT8	0.8879	0.8881	0.6786	112.02

Overall, the ablation studies confirm that multimodal alignment, negative sampling, and optimized stride are necessary for robust classification, while quantization offers significant deployment benefits with manageable accuracy trade-offs.

3.4 Computational Efficiency and Edge Deployment

The efficiency and deployability of TinyCLIP were evaluated on two hardware platforms: (1) NVIDIA T4 GPU (cloud environment), and (2) NVIDIA Jetson Nano (edge deployment).

Performance metrics include average patch latency, full-image inference time, throughput, and memory usage. Tables 5 and 6 summarize the results.

TABLE 5: FP16 TensorRT Deployment Performance

Metric	Value
Engine Size	126.98 MB
Average Latency	10.38 ms
Throughput	96.33 FPS
Peak Memory (CUDA)	6452.11 MB
PyTorch Memory	9.28 MB

FP16 inference processed approximately 96 patches per second on the NVIDIA T4 GPU. This value represents patch-level throughput and should not be interpreted as the number of complete orchard images processed per second. Because each full-resolution image was decomposed into multiple overlapping patches, processing all patches and aggregating their predictions into class-specific heatmaps required approximately 6 seconds per full image. Thus, the reported throughput characterizes computational efficiency at the patch level, whereas the end-to-end full-image inference rate was approximately 0.17 images per second. GPU memory usage remained within the capacity of the desktop hardware configuration.

TABLE 6: INT8 TensorRT Deployment Performance

Metric	Value
Engine Size	137.15 MB
Average Latency	8.93 ms
Throughput	112.02 FPS
Peak Memory (CUDA)	6519.40 MB
PyTorch Memory	9.28 MB

INT8 inference further improved speed to 112 FPS, demonstrating the benefit of aggressive quantization. However, this came at a measurable reduction in anatomical recall, particularly for thin peduncles. This trade-off is acceptable for fast orchard scanning but should be considered carefully for precision thinning tasks.

On the Jetson Nano, batches of eight patches achieved successful inference within a 5–6 s timeframe per orchard image. This ensures compatibility with ground vehicles or handheld thinning devices that require rapid anatomical awareness.

Overall Deployment Conclusions TinyCLIP, with TensorRT optimization, satisfies real-world latency, memory, and throughput requirements for field deployment. FP16 offers the best balance between accuracy and speed, while INT8 enables ultra-fast scanning with modest accuracy loss. These benchmarks validate the feasibility of integrating lightweight VLMs into orchard robots for early-season management.

3.5 Peduncle Identification for Autonomous Fruit Thinning

Accurate peduncle localization is a key perceptual requirement for robotic fruitlet thinning in commercial apple orchards. While detecting fruitlets indicates the presence of a potential thinning target, the peduncle represents the actual cutting point for scissor-type robotic end-effectors. Therefore, identifying peduncles reliably within complex canopy environments is essential for enabling safe and

Fig. 10: Representative examples of correctly classified peduncle patches extracted from an early-season orchard image. The figure highlights the model's ability to identify slender, low-contrast peduncles under challenging conditions including occlusion, variable illumination, and canopy clutter. These examples demonstrate the effectiveness of the lightweight TinyCLIP vision–language model in recognizing anatomically meaningful peduncle structures essential for autonomous robotic fruit thinning.

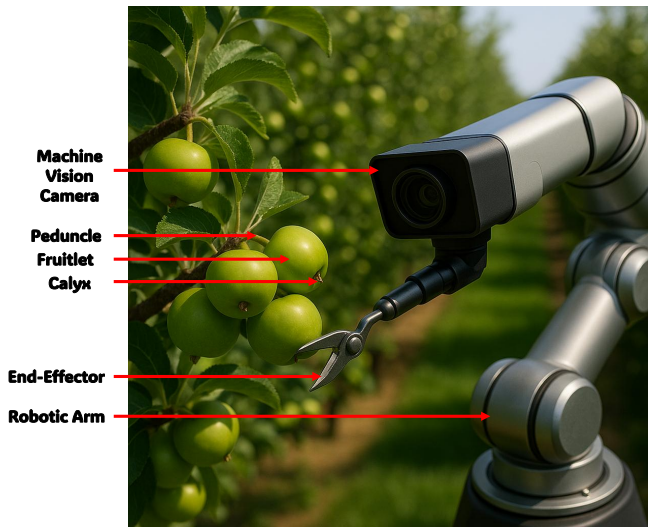


Fig. 11: Conceptual future roadmap for autonomous green-fruit thinning. A robotic arm equipped with a multimodal machine-vision camera and precision cutting end-effector identifies fruitlet anatomical structures—calyx, fruitlet body, and peduncle within a dense orchard canopy. The system illustrates how lightweight vision-language models can guide peduncle-targeted removal for safe, efficient, and scalable robotic thinning in commercial orchards.

precise automated thinning when specific types of end-effectors are used. Figure 10 illustrates representative examples of correctly classified peduncle patches identified by the proposed TinyCLIP-based multimodal framework in early-season orchard images.

Peduncles present several challenges for automated detection. They appear as slender, elongated structures with fine surface trichomes that often visually blend with surrounding stems and foliage. Their appearance is further complicated by variations in illumination, occlusion by leaves, and differences in orientation within dense fruit clusters. These factors introduce substantial intra-class variability, making peduncle identification difficult for conventional visual detectors relying solely on pixel-level features. Despite these challenges, the correctly classified patches shown in Figure 10 demonstrate that the adapted TinyCLIP model can learn stable visual-semantic cues associated with peduncle morphology.

A key customization of TinyCLIP for the apple thinning problem lies in the integration of domain-specific language prompts and patch-based anatomical classification. By using prompts such as "a photo of a peduncle," the multimodal framework guides the visual encoder toward semantically meaningful plant structures rather than generic object categories. Combined with the sliding-window patch analysis strategy, this approach enables the model to detect thin peduncle structures that might otherwise be overlooked in full-image analysis. The multimodal alignment between image patches and horticultural terminology helps distinguish peduncles from visually similar background elements such as stems or leaf veins.

From an operational perspective, reliable peduncle identification directly affects the success of autonomous thinning systems when pendiuncle cutting end-effectors are

used. In that case, a robotic platform must first identify candidate fruitlets and then determine the precise peduncle location to execute a targeted removal without damaging neighboring fruitlets or buds. Misidentification can lead to incorrect cutting positions, incomplete removal, or unintended damage within the fruit cluster. The results shown in Figure 10 indicate that the proposed TinyCLIP-based perception pipeline can robustly isolate peduncle structures even under challenging orchard conditions including foliage occlusion, low contrast, and diverse peduncle orientations. These capabilities provide critical anatomical cues required for downstream robotic manipulation.

Looking ahead, Figure 11 outlines the broader robotic framework enabled by this work that utiluzes a sccissor-type end-effector, which has shown to be one of the most effective one through our unpublished work on fruitlet thinning end-effector design. It is envisioned that an autonomous robotic arm equipped with a lightweight multimodal perception system analyzes orchard scenes to identify fruitlet anatomical components and localize peduncles as actionable cutting targets. The adapted TinyCLIP model provides semantic understanding of orchard structures by combining visual cues with horticultural language prompts. Patch-level predictions can be aggregated into spatial probability maps, which can then be fused with depth sensing to guide robotic motion planning. The robotic end-effector can align with the identified peduncle and perform a precise removal action before navigating to the next fruitlet cluster.

This framework represents a shift from traditional feature-engineered detection pipelines toward semantically grounded multimodal perception tailored for agricultural robotics. By customizing TinyCLIP with domain-specific prompts, patch-based anatomical analysis, and deployment-oriented optimization, the proposed system demonstrates how lightweight vision–language models can support practical robotic thinning operations in real orchard environments. Together, Figures 10 and 11 highlight the potential of multimodal perception combined with robotic manipulation to enable scalable and autonomous crop-load management in commercial apple orchards.

4 Conclusion

This study demonstrates that a lightweight multimodal vision–language architecture, TinyCLIP, can effectively address one of the most challenging perception tasks in precision horticulture: fine-grained classification and localization of early-stage apple fruitlet anatomy under complex orchard conditions. By combining patch-based multi-label learning with text-guided semantic alignment, the proposed framework successfully identifies calyxes, fruitlets, peduncles, and background regions despite strong occlusion, minimal color contrast, and highly cluttered canopies. The sliding-window inference mechanism further enables the generation of continuous heatmaps, offering interpretable spatial cues that support downstream robotic tasks such as peduncle-aware fruitlet thinning. Extensive experiments across cloud GPUs and edge hardware show that TinyCLIP maintains strong discriminative performance even after aggressive quantization. FP16 and INT8 deployments on the NVIDIA Jetson Orin preserve high F1-scores, and the optimized TensorRT pipeline achieves rapid inference speeds suitable for real-time field deployment. This efficiency, paired with the model's multimodal grounding, provides a practical bridge between high-capacity VLM reasoning and the resource-constrained demands of agricultural robotics. Beyond classification accuracy, the study highlights the significance of anatomically meaningful localization for robotic thinning. Reliable peduncle detection, in particular, has direct operational consequences for safe and effective fruit removal. While heatmaps do not provide instance-level segmentation or exact counting, they offer robust presence and spatial likelihood information essential for early-season decision-making. Overall, this work establishes TinyCLIP as an interpretable, lightweight, and deployable multimodal solution for orchard perception, laying the foundation for scalable robotic thinning, automated crop-load assessment, and next-generation intelligent orchard systems. Future extensions may integrate grounding-based VLMs or multimodal fusion frameworks to further enhance fine-grained localization and cluster-level reasoning.

Acknowledgement

This work was supported in part by the National Science Foundation (NSF); in part by United States Department of Agriculture (USDA); in part by the National Institute of Food and Agriculture (NIFA), through the "Artificial Intelligence (AI) Institute for Agriculture" Program, Accession Num ber 1029004 for the Project Titled "Robotic Blossom Thinning with Soft Manipulators" under Award AWD003473, Award AWD004595, and Award USDA-NIFA; and in part by United States Department of Agriculture National Science Foundation (USDANSF), Accession Number 1031712, under the Project "ExPanding University of Central Florida (UCF) AI Research To Novel Agricultural EngineeRing Applications (PARTNER)" under Grant 2024-67022-41788. Additionally, this work was supported in part by the Intramural Research Program of the U.S. Department of Agriculture (USDA), National Institute of Food and Agriculture, under Grant No. 2024-67022-41788. The views and conclusions expressed in this paper are those of the authors and do not necessarily reflect the official policies or positions of the USDA or the U.S. Government.

Declarations

The authors declare no conflicts of interest.

Statement on AI Writing Assistance

ChatGPT and Perplexity were utilized to enhance grammatical accuracy and refine sentence structure; all AI-generated revisions were thoroughly reviewed and edited for relevance. Additionally, ChatGPT-4o was employed to generate realistic visualization in Figure 11.

References

[1] G. Costa, A. Botton, and G. Vizzotto, "Fruit thinning: Advances and trends," *Horticultural reviews*, vol. 46, pp. 185–226, 2018.

[2] D. Greene and G. Costa, "Fruit thinning in pome-and stone-fruit: State of the art," in *EUFRIN Thinning Working Group Symposia 998*, pp. 93–102, 2012.

[3] G. Ouma, "Fruit thinning with specific reference to citrus species: A review," *Agric. Biol. JN Am*, vol. 3, no. 4, pp. 175–191, 2012.

[4] P. M. Hirst *et al.*, "Advances in understanding flowering and pollination in apple trees," *Achieving sustainable cultivation of apples*, pp. 109–126, 2017.

[5] K. Davis, E. Stover, and F. Wirth, "Economics of fruit thinning: A review focusing on apple and citrus," *HortTechnology.*, vol. 14, no. 2, p. 282, 2004.

[6] M. Goffinet, T. Robinson, and A. Lakso, "A comparison of 'empire'apple fruit size and anatomy in unthinned and hand-thinned trees," *Journal of Horticultural Science*, vol. 70, no. 3, pp. 375–387, 1995.

[7] R. S. Sidhu, S. A. Bound, and I. Hunt, "Crop load and thinning methods impact yield, nutrient content, fruit quality, and physiological disorders in 'scilate'apples," *Agronomy*, vol. 12, no. 9, p. 1989, 2022.

[8] L. Prause, "Digital agriculture and labor: A few challenges for social sustainability," *Sustainability*, vol. 13, no. 11, p. 5980, 2021.

[9] B. Sims and J. Kienzle, "Sustainable agricultural mechanization for smallholders: what is it and how can we implement it?," *Agriculture*, vol. 7, no. 6, p. 50, 2017.

[10] P. Callea, G. Zimbalatti, E. Quendler, A. Nimmerichter, N. Bachl, B. Bernardi, D. Smorto, and S. Benalia, "Occupational illnesses related to physical strains in apple harvesting," *Annals of Agricultural and Environmental Medicine*, vol. 21, no. 2, 2014.

[11] F. A. Fathallah, "Musculoskeletal disorders in labor-intensive agriculture," *Applied ergonomics*, vol. 41, no. 6, pp. 738–743, 2010.

[12] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[13] J. E. Hoffmann, H. G. Tosso, M. M. D. Santos, J. F. Justo, A. W. Malik, and A. U. Rahman, "Real-time adaptive object detection and tracking for autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 450–459, 2020.

[14] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 502–511, 2019.

[15] Z. Zhou, L. Li, A. Fürsterling, H. J. Durocher, J. Mouridsen, and X. Zhang, "Learning-based object detection and localization for a mobile robot manipulator in sme production," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102229, 2022.

[16] X. Shi, S. Wang, B. Zhang, X. Ding, P. Qi, H. Qu, N. Li, J. Wu, and H. Yang, "Advances in object detection and localization techniques for fruit harvesting robots," *Agronomy*, vol. 15, no. 1, p. 145, 2025.

[17] P. K. Mishra and G. Saroha, "A study on video surveillance system for object detection and tracking," in *2016 3rd international conference on computing for sustainable global development (INDIACom)*, pp. 221–226, IEEE, 2016.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, Ieee, 2005.

[19] Z. Chen, K. Chen, and J. Chen, "Vehicle and pedestrian detection using support vector machine and histogram of oriented gradients features," in *2013 International Conference on Computer Sciences and Applications*, pp. 365–368, IEEE, 2013.

[20] B. Sugiarto, E. Prakasa, R. Wardoyo, R. Damayanti, L. M. Dewi, H. F. Pardede, Y. Rianto, *et al.*, "Wood identification based on histogram of oriented gradient (hog) feature and support vector machine (svm) classifier," in *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 337–341, IEEE, 2017.

[21] L. Rosyidi, A. Prasetyo, and M. S. Romadhon, "Object tracking with raspberry pi using histogram of oriented gradients (hog) and support vector machine (svm)," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*, pp. 1–6, IEEE, 2020.

[22] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[23] X. Zou, "A review of object detection techniques," in *2019 International conference on smart grid and electrical automation (ICSGEA)*, pp. 251–254, IEEE, 2019.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[25] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[29] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6569–6578, 2019.

[30] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, 2020.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[32] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, 2018.

[33] R. Sapkota, M. Flores-Calero, R. Qureshi, C. Badgujar, U. Nepal, A. Poulose, P. Zeno, U. B. P. Vaddevolu, S. Khan, M. Shoman, *et al.*, "Yolo advances to its genesis: a decadal and comprehensive review of the you only look once (yolo) series," *Artificial Intelligence Review*, vol. 58, no. 9, p. 274, 2025.

[34] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine learning and knowledge extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[35] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[36] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[37] M. Jamali, P. Davidsson, R. Khoshkangini, M. G. Ljungqvist, and R.-C. Mihailescu, "Context in object detection: a systematic literature review," *Artificial Intelligence Review*, vol. 58, no. 6, pp. 1–89, 2025.

[38] A. Wang, J. Li, and L. Pan, "Trifusenet: Open-vocabulary sign language translation via hierarchical image-text-keypoint fusion," *Journal of Circuits, Systems and Computers*, p. 2550375, 2025.

[39] H. Shahmohammadi, M. Heitmeier, E. Shafaei-Bajestan, H. P. Lensch, and R. H. Baayen, "Language with vision: A study on grounded word and sentence embeddings," *Behavior Research Methods*, vol. 56, no. 6, pp. 5622–5646, 2024.

[40] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia, "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.

[41] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PmLR, 2021.

[42] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, "Reproducible scaling laws for contrastive language-image learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2818–2829, 2023.

[43] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International conference on machine learning*, pp. 12888–12900, PMLR, 2022.

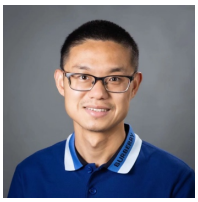
[44] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M. Rohrbach, and D. Kiela, "Flava: A foundational language and vision alignment model," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15638–15650, 2022.

[45] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, *et al.*, "Grounded language-image

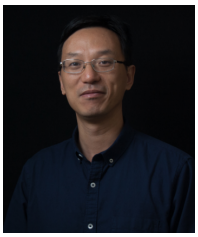
pre-training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10965–10975, 2022.
[46] T. Ren, Q. Jiang, S. Liu, Z. Zeng, W. Liu, H. Gao, H. Huang, Z. Ma, X. Jiang, Y. Chen, *et al.*, "Grounding dino 1.5: Advance the" edge" of open-set object detection," *arXiv preprint arXiv:2405.10300*, 2024.
[47] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European conference on computer vision*, pp. 38–55, Springer, 2024.
[48] S. Wang, D. Kim, A. Taalimi, C. Sun, and W. Kuo, "Learning visual grounding from generative vision and language model," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 8057–8067, IEEE, 2025.
[49] A.-C. Cheng, H. Yin, Y. Fu, Q. Guo, R. Yang, J. Kautz, X. Wang, and S. Liu, "Spatialrgpt: Grounded spatial reasoning in vision-language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 135062–135093, 2024.
[50] W. Xu, T. Zhou, T. Zhang, J. Li, P. Chen, J. Pan, and X. Liu, "Exploring grounding abilities in vision-language models through contextual perception," *IEEE Transactions on Cognitive and Developmental Systems*, 2025.
[51] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 46, no. 8, pp. 5625–5644, 2024.
[52] G. Luo, Y. Zhou, T. Ren, S. Chen, X. Sun, and R. Ji, "Cheap and quick: Efficient vision-language instruction tuning for large language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 29615–29627, 2023.
[53] X. Li, C. Wen, Y. Hu, Z. Yuan, and X. X. Zhu, "Vision-language models in remote sensing: Current progress and future trends," *IEEE Geoscience and Remote Sensing Magazine*, vol. 12, no. 2, pp. 32–66, 2024.
[54] K. Wu, H. Peng, Z. Zhou, B. Xiao, M. Liu, L. Yuan, H. Xuan, M. Valenzuela, X. S. Chen, X. Wang, *et al.*, "Tinyclip: Clip distillation via affinity mimicking and weight inheritance," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21970–21980, 2023.
[55] P. K. A. Vasu, H. Pouransari, F. Faghri, R. Vemulapalli, and O. Tuzel, "Mobileclip: Fast image-text models through multi-modal reinforced training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15963–15974, 2024.
[56] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "Eva-clip: Improved training techniques for clip at scale," *arXiv preprint arXiv:2303.15389*, 2023.
[57] Y. Mi, Y. Li, Y. Li, C. Hui, T. Zhang, Z. Li, C. Song, W. Y. B. Lim, and S. Liu, "Q-clip: Unleashing the power of vision-language models for video quality assessment through unified cross-modal adaptation," *arXiv preprint arXiv:2508.06092*, 2025.

Ranjan Sapkota (Member, IEEE) is a Ph.D. student at Cornell University in the Department of Biological and Environmental Engineering. His research centers on artificial intelligence and robotics in agriculture, with expertise spanning automation systems, machine vision, robot manipulation, multimodal large language models (MM-LLMs), deep learning, agentic AI and generative AI technologies. From 2022 to 2024, he was with Washington State University, advancing research in agricultural automation and intelligent robotic systems. He previously earned his M.S. in Agricultural and Biosystems Engineering from North Dakota State University, USA (2020–2022), where he specialized in computer vision, GIS, remote sensing, agricultural machinery, and UAV applications for agriculture.

William Bu graduated with a Bachelor of Science in Computer Science, with honors, from the University of Central Florida in 2026. His work spans computer vision, applied machine learning, and full-stack software development, including research experience in AI-driven agricultural applications and hands-on projects involving object detection, real-time systems, and AI-powered analytics tools. He is interested in building intelligent, practical systems at the intersection of software engineering and artificial intelligence.



Dr. Chen Chen is an Associate Professor at the Institute of Artificial Intelligence (IAI) at the University of Central Florida. His research spans computer vision, multimodal and efficient deep learning, federated learning, and medical image computing, with broad applications in healthcare, sensing, and intelligent systems. His recent work focuses on federated and privacy-preserving learning frameworks with potential for healthcare deployment, and on multimodal foundation models that integrate visual and language understanding across domains.



Dr. Yunjun Xu received the Ph.D. degree in aerospace engineering from the University of Florida in 2003. Currently, he is a Professor with the Department of Mechanical and Aerospace Engineering, University of Central Florida. His current research interests include AI based modeling and optimization, control theory, and field robotics.

Prof. Dr. Manoj Karkee is the Norman R. and Sharon R. Scott Professor of Agriculture and Life Sciences at the Department of Biological and Environmental Engineering Department at Cornell University. He received his PhD in Agricultural Engineering and Human Computer Interaction from Iowa State University and has been director and professor at Washington State University Center for Precision and Automated Agricultural Systems. Dr. Karkee leads a strong research and education program in the area of sensing, machine vision, AI, and Robotics in Agriculture. He has published widely in such journals as 'Computers and Electronics in Agriculture', 'Computers in Industry', 'Journal of Field Robotics', and 'Journal of the American Society of Agricultural and Biological Engineers (ASABE)', and has been an invited speaker at numerous national and international conferences and universities. Dr. Karkee is currently serving as the Editor-in-Chief for 'Computers and Electronics in Agriculture', and associate editor of 'Journal of the ASABE' and has served as a guest editor for 'Journal of Field Robotics'. He is also an elected chair of CIGR (International Commission of Agricultural and Biosystems Engineering) Section III - Plant Production, and IFAC (International Federation of Automatic Control) Technical Committee 8.1 - Control in Agriculture. Dr. Karkee was awarded '2020 Rainbird Engineering Concept of the Year' by ASABE, and was recognized as '2019 Pioneer in Artificial Intelligence and IoT' by Connected World magazine.