2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

CropNeRF: A Neural Radiance Field-Based Framework for Crop Counting

Md Ahmed Al Muzaddid and William J. Beksi

***Abstract*— Rigorous crop counting is crucial for effective agricultural management and informed intervention strategies. However, in outdoor field environments, partial occlusions combined with inherent ambiguity in distinguishing clustered crops from individual viewpoints poses an immense challenge for image-based segmentation methods. To address these problems, we introduce a novel crop counting framework designed for exact enumeration via 3D instance segmentation. Our approach utilizes 2D images captured from multiple viewpoints and associates independent instance masks for neural radiance field (NeRF) view synthesis. We introduce crop visibility and mask consistency scores, which are incorporated alongside 3D information from a NeRF model. This results in an effective segmentation of crop instances in 3D and highly-accurate crop counts. Furthermore, our method eliminates the dependence on crop-specific parameter tuning. We validate our framework on three agricultural datasets consisting of cotton bolls, apples, and pears, and demonstrate consistent counting performance despite major variations in crop color, shape, and size. A comparative analysis against the state of the art highlights superior performance on crop counting tasks. Lastly, we contribute a cotton plant dataset to advance further research on this topic.**

***Index Terms*— Agricultural Automation; Computer Vision for Automation; Object Detection, Segmentation, Categorization**

I. Introduction

With a declining migrant workforce and a growing global population, the agricultural sector faces significant challenges that demand innovative solutions. Moreover, the adverse effects of climate change, such as unpredictable weather patterns and water scarcity, necessitate the creation of more adaptive and efficient farming practices. By leveraging advanced technologies (e.g., robotic automation, computer vision, sensor networks, etc.) to improve resource efficiency and optimize productivity, precision agriculture is a practical and effective response to these complications.

Accurate crop counting is essential for resource optimization, yield estimation, and post-harvest management in precision agriculture. Nonetheless, counting crops in field conditions remains problematic. For example, occlusions caused by foliage, branches, and neighboring plants complicates crop detection and re-identification thus increasing the likelihood of double counting. In addition, crop overlap and clustering hinder the precise differentiation of individual instances, while variations in color, shape, and size across growth stages further impede the counting process.

Various approaches have been proposed to address these challenges [1], yet very few leverage the 3D structure of the problem. In this paper, we introduce a novel approach to instance segmentation that effectively distinguishes individual instances within a reconstructed 3D environment, thereby providing an accurate enumeration of the target crop. Our method demonstrates resilience to inaccuracies in the underlying 2D instance segmentation masks upon which it depends. Additionally, it exploits volumetric information to handle issues such as occlusion and perspective-induced ambiguity, and can be deployed on crops of different physical qualities minus the need for recalibration.

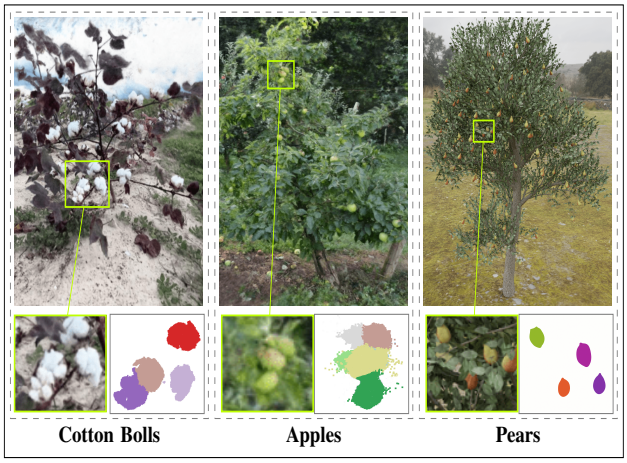

Fig. 1: An overview of crop counting via 3D instance segmentation. The zoomed-in views depict clusters of cotton bolls, apples, and pears from a cotton plant, apple tree, and pear tree, respectively. The instance segmentation results for these clusters, obtained using CropNeRF, are shown in color on the right-hand side of the zoomed-in views.

This work presents a crop neural radiance field (CropNeRF) 3D instance segmentation framework that accurately differentiates individual instances of target crops, Fig. 1. By integrating visibility and instance mask consistency scores, CropNeRF effectively addresses occlusions and ambiguities caused by changes in perspective while adapting to crops of varying dimensions without requiring sensitive parameter tuning. Unlike existing instance segmentation approaches (e.g., [2], [3]), which depend on establishing correspondences between instance masks across multiple images, CropNeRF is designed to operate independently on each mask. This eliminates the need for explicit inter-image mapping, thereby simplifying the 3D segmentation process while improving accuracy.

Given a set of images from multiple viewpoints along with the corresponding camera poses and 2D instance masks of the target crop, CropNeRF accurately counts instances

The authors are with the Department of Computer Science and Engineering, The University of Texas at Arlington, Arlington, TX, USA. Emails: mdahmedal.muzaddid@mavs.uta.edu, william.beksi@uta.edu.

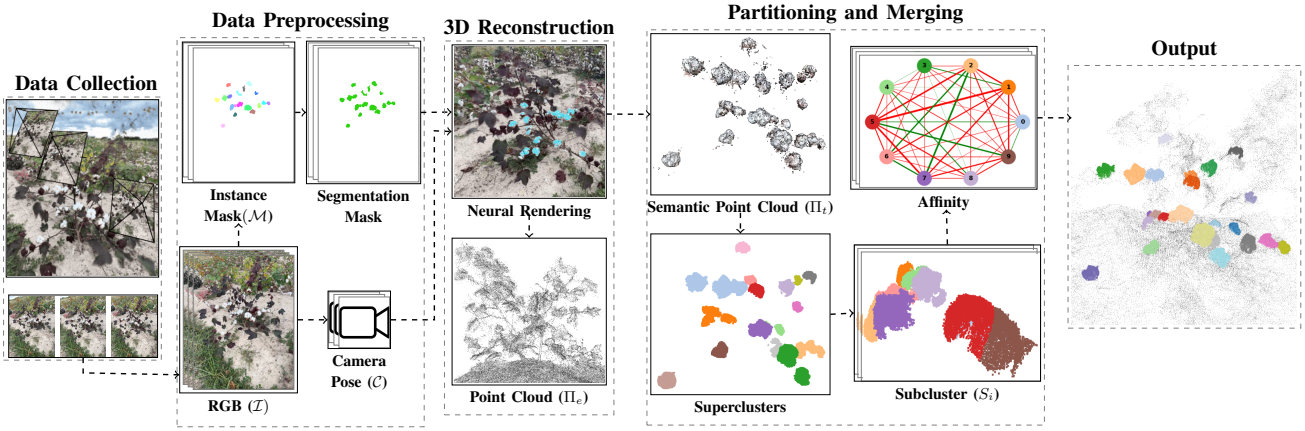

Fig. 2: The crop counting pipeline. Multiple images of the target crop are captured from different viewpoints during the data collection step. In the data preprocessing stage, camera poses ($\mathcal{C}$) are extracted from the captured images ($\mathcal{I}$), and the crops are segmented into instance masks ($\mathcal{M}$). Semantic segmentation masks are then generated from the instance masks. The extracted camera poses, captured images, and semantic masks are used to train a semantic NeRF model during the 3D reconstruction stage. The trained model generates a semantic point cloud representing the crops (Π_t) and a point cloud representing the environment (Π_e). In the partitioning and merging step, the semantic point cloud is first divided into superclusters and then into subclusters (S_i). The affinity among subclusters is calculated based on subcluster visibility and mask consistency scores. Lastly, the subclusters are merged based on subcluster affinity to produce the final output.

of the crop via 3D instance segmentation. To do this, we learn a high-fidelity 3D reconstruction while simultaneously modeling a semantic field. From the semantic field, we extract a point cloud representing only the targeted crop, which is then partitioned into subclusters. The visibility and mask consistency of the subclusters is computed based on the camera poses, corresponding point clouds, and associated instance masks. These scores are then aggregated across multiple views, to merge the subclusters into distinct instances of the target crop, allowing for exact crop counting.

In summary, we make the following contributions.

- We develop a 3D instance segmentation method capable of differentiating crops of varying colors, shapes, and sizes.
- We introduce a mask reliability score that incorporates crop visibility and mask consistency, enabling robustness against occlusions and annotation discrepancies.
- We release a public infield cotton plant dataset designed for 3D rendering and cotton boll counting tasks.

The source code, dataset, and multimedia material associated with this project can be found at https://robotic-vision-lab.github.io/cropnerf.

II. Related Work

A. Image-Based Techniques

Image-based methods typically employ object detection to identify crops within images. For example, Chen et al. [4] utilized multiple convolutional neural networks (CNNs) to map input images to total fruit counts. Similarly, Häni et al. [5] formulated crop counting as a multi-class classification problem where a CNN predicts a single count per detected crop cluster. Tedesco et al. [6] applied detection models (e.g., Faster R-CNN [7], SSD [8], and MobileNetV2 [9]) and selected architectures based on computational constraints. Likewise, James et al. [10], [11] evaluated object detection models along with transfer learning to classify citrus fruit on trees versus on the ground for yield estimation. While modern object detection models can achieve high accuracy under outdoor lighting conditions, performance is often adversely affected by occlusions due to limited perspective and the absence of depth information.

B. Video-Based Tracking Methods

To handle occlusions and facilitate farm-level crop counting, previous works have implemented multi-frame techniques based on a tracking-by-detection paradigm. For instance, Smitt et al. [12] counted sweet peppers in RGBD videos by applying Mask R-CNN [13] for segmentation along with wheel odometry and depth information to enhance tracking accuracy. Zhang et al. [14] modified YOLOv3 [15] to improve the detection of small fruits and implemented a region-based counting strategy for tracking. Al Muzaddid and Beksi [16] created a cotton boll counting framework that leverages the spatial relationships among neighboring bolls to re-identify occluded instances. In proceeding work, Al Muzaddid et al. [17] developed a crop tracking approach based on the combination of appearance and motion information. Although these methods are effective, their performance is dependent on the accuracy of the object detectors and trackers.

C. 3D Reconstruction-Based Solutions

Structure-from-motion (SfM) has been employed to generate 3D point cloud reconstructions of orchards for precise crop localization. A cotton boll counting method by Sun et al. [18] made use of 3D point clouds reconstructed from multi-view field images using SfM. Matos et al. [19] proposed a frame-to-frame tracking technique for fruit counting across image sequences by leveraging 3D information derived from SfM. Nonetheless, these approaches often require extensive

camera calibration and may produce noisy reconstructions in environments with dense foliage.

NeRFs [20] are a popular methodology for 3D reconstruction. They offer high-fidelity volumetric representations without the need for extensive camera calibration. For example, FruitNeRF [21] is a NeRF-based fruit counting framework. It relies on clustering-based segmentation, which assumes consistent crop sizes and requires manual parameter tuning. This limits the generalizability of FruitNeRF across different crop growth stages. In contrast, our approach eliminates the need for delicate parameter tuning based on crop size and distribution.

3D Gaussian splatting (3DGS) has emerged as alternative to NeRFs. Jiang et al. [22] used 3DGS to reconstruct high-fidelity 3D models of cotton plants. However, the dataset was collected indoors, under controlled lighting conditions, and includes only defoliated cotton plants. Zhang et al. [23] applied 3DGS to reconstruct wheat plots and extract morphological traits (e.g., head length, width, and volume) via segmentation. Yet, the segmentation technique relies on an iterative process based on the overlap between projected 3D Gaussians and 2D segmentation masks, quantified using the intersection over union and filtered by empirical thresholds. Differently, our method accounts for crop visibility when evaluating mask overlap and eliminates the need for arbitrary thresholding. The end result is a more robust, crop-agnostic segmentation framework suitable for field conditions.

III. Neural Radiance Field-Based Framework for Crop Counting

Let $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ be a set of RGB images that capture the target environment from multiple viewpoints, where $I_i \in \mathbb{R}^{h \times w \times 3}$ and each image is of resolution $h \times w$. In addition, assume that for each image an instance mask $\mathcal{M} = \{M_1, M_2, \ldots, M_n\}$ of the target crop is available and $M_j \in \mathbb{R}^{h \times w}$. We proceed by extracting camera poses $\mathcal{C} = \{C_1, C_2, \ldots, C_n\} \in \mathbb{R}^{3 \times 4}$ from the set of images. For consistency, we use C_j and M_j to denote the camera pose and instance mask, respectively, associated with image I_j. Note that C_j may be used interchangeably to refer to the j^{th} camera, its pose, or the associated camera view where the context clarifies the intended meaning. An overview of the crop counting pipeline is illustrated in Fig. 2.

A. Volumetric Rendering

Given the images $\mathcal{I}$ and their corresponding camera poses $\mathcal{C}$, we train a NeRF model to reconstruct a high-fidelity 3D representation of the environment. The model encodes the scene within a multilayer perceptron via mapping a position $x = (x, y, z)$ and a viewing direction $d = (\phi, \theta)$ to a volume density σ and RGB radiance $c = (r, g, b)$. The density field $\mathcal{F}_\sigma : x \rightarrow \sigma$ is defined as a function of the position, while the appearance field $\mathcal{F}_c : (x, d) \rightarrow c$ depends on both the position and the viewing direction. Building upon the concept of semantic fields [24], we augment the NeRF framework to integrate crop-specific information. Concretely, we utilize a set of masks $\mathcal{M}$ to embed crop information into the 3D volume through a semantic field $\mathcal{F}_s : x \rightarrow s$ that predicts the spatial logits.

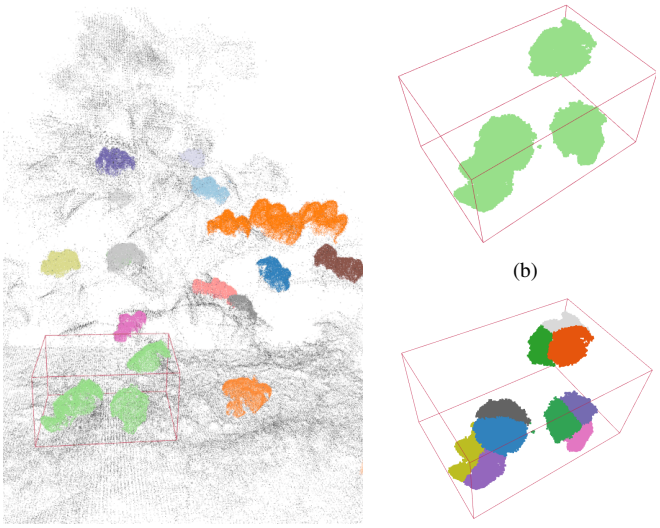

Fig. 3: The point cloud clustering process: (a) point cloud Π_t representing the crops is segmented into superclusters, each identified by a unique color; (b) a single supercluster visualized in 3D; (c) the supercluster is further partitioned into multiple subclusters.

B. Point Cloud Generation

The trained NeRF model encodes the spatial information of the entire scene (i.e., soil, trunk, stem, foliage, and crop) within the density field $\mathcal{F}_\sigma$. Conversely, the semantic field $\mathcal{F}_s$ exclusively represents the spatial distribution of the target crop within the 3D volume. From these two fields, we generate two separate point clouds, one representing the environment (including the target crop) and another one representing only the target crop, by sampling $\mathcal{F}_\sigma$ and $\mathcal{F}_s$, respectively. To extract the point cloud representing the environment Π_e, we uniformly sample points from $\mathcal{F}_\sigma$. Nevertheless, extracting points that exclusively represent the target crop Π_t by sampling $\mathcal{F}_s$ leads to a sparse point cloud. To address this issue, we follow the approach used in FruitNeRF where sampled semantic points are filtered based on their corresponding density values resulting in a denser representation.

C. Point Cloud Preprocessing

We preprocess the point cloud Π_t by partitioning it using a two-phase approach. In the first phase, Π_t is segmented into superclusters, $\psi_1, \psi_2, \ldots, \psi_\kappa$, such that $\Pi_t = \dot{\bigcup}_{k=1}^{\kappa} \psi_k$ where $\dot{\bigcup}$ denotes the disjoint set union (Fig. 3a). The goal of this step is to separate spatially distant and independent crop instances, thereby decomposing the problem into smaller and more manageable subproblems that can be efficiently addressed. We employ DBSCAN [25] to identify the superclusters based on point density without requiring prior assumptions about the number or size of the clusters.

We refine the extracted superclusters by removing outliers based on the clustering results, ensuring a more coherent

segmentation (Fig. 3b). It is important to observe that the output of our method remains invariant to the number of superclusters. For example, Π_t can be considered as a single supercluster without affecting the final segmentation results. However, to improve computational efficiency and minimize unnecessary interactions between independent regions of the point cloud, we incorporate the first-phase clustering as a preprocessing step. Each supercluster is segmented independently and the results from all superclusters are aggregated to generate the output.

In the second phase, we further divide a supercluster ψ_i into K subclusters. This is done using the k-means clustering algorithm, depicted in Fig. 3c, such that $\psi_i = \dot{\bigcup}_{k \in K} S_k$. The objective of this step is to partition a supercluster into smaller subclusters, guaranteeing that each crop instance comprises at least one or more subclusters. We achieve this by decreasing the size of each cluster through the selection of a larger K value. Note that for a moderately large K, the segmentation results remain robust to variations in K as discussed in the evaluation (Sec. IV-G).

D. Subcluster Visibility

Due to occlusions, some subclusters may only be partially observable from certain viewpoints. To quantify their visibility, we assign a score $v_{ij} \in [0, 1]$ to each subcluster S_i as viewed from camera $\mathcal{C}_j$. More formally,

$$v_{ij} = \frac{Area\ of\ S_i\ visible\ from\ \mathcal{C}_j}{Area\ of\ S_i\ visible\ from\ \mathcal{C}_j\ without\ occlusions} = \frac{Area\left(\mathcal{V}_j\left(S_i\right)\right)}{Area\left(\mathcal{P}_j\left(S_i\right)\right)}. \tag{1}$$

In (1), $\mathcal{P}_j(S_i)$ is a function that projects point cloud S_i onto the camera plane using the camera matrix, which is the product of the intrinsic and extrinsic matrices derived from $\mathcal{C}_j$. We refer to this as the occlusion-free projection of S_i. The occlusion-aware projection function is represented by $\mathcal{V}_j(S_i)$ and accounts for environmental elements such as the stems, foliage, and crops (i.e., any objects between the camera $\mathcal{C}_j$ and S_i that may occlude S_i). We obtain $\mathcal{V}_j(S_i)$ by projecting both S_i and Π_e onto the camera plane while incorporating depth information to account for occlusions.

E. Mask Consistency

The instance masks $\mathcal{M}$ are based on 2D images and therefore inherently susceptible to inconsistencies, regardless of whether they are annotated by a machine learning model or a human. These deviations arise from the fixed camera viewpoint and the absence of depth information, hindering precise differentiation of closely-spaced instances. To account for variations in the masks, we define a consistency score $c_{ij} \in [0, 1]$. This score quantitatively evaluates the consistency of the mask M_j that overlaps with $\mathcal{V}_j\left(S_i\right)$, the occlusion-aware projection of S_i. The consistency score is computed as

$$c_{ij} = \frac{\max_l Area\left(\mathcal{V}_j\left(S_i\right) \cap M_{jl}\right)}{Area\left(\mathcal{V}_j\left(S_i\right)\right)}, \tag{2}$$

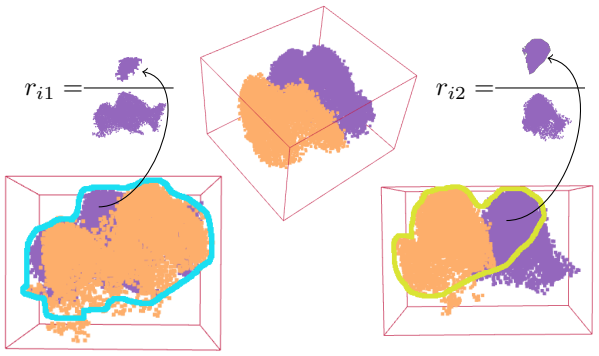

Fig. 4: A visual representation of computing the mask reliability score. The score of the purple subcluster is calculated based on two different camera views. Only the visible projection area of the subcluster that overlaps with the masks (represented by the cyan and yellow boundaries) is considered in the numerator.

where M_{jl} denotes the region of M_j labeled as instance l. The max operation ensures that if multiple distinct labels in M_j overlap with $\mathcal{V}_j\left(S_i\right)$, then only the region with label l that has maximum overlap with $\mathcal{V}_j\left(S_i\right)$ is considered. We also assign the label,

$$\lambda_{ij} = \underset{l}{\operatorname{argmax}}\, Area\left(\mathcal{V}_j\left(S_i\right) \cap M_{jl}\right), \tag{3}$$

to subcluster S_i in M_j.

F. Mask Reliability

To quantify the confidence level of an instance mask associated with subcluster S_i in view M_j, we derive a reliability score $r_{ij} \in [0, 1]$. As illustrated in Fig. 4, r_{ij} is computed by combining the subcluster visibility score v_{ij} and the mask consistency score c_{ij}. Concretely,

$$\begin{aligned} r_{ij} &= v_{ij} c_{ij} \\ &= \frac{Area\left(\mathcal{V}_j\left(S_i\right)\right)}{Area\left(\mathcal{P}_j\left(S_i\right)\right)} \cdot \frac{\max_l Area\left(\mathcal{V}_j\left(S_i\right) \cap M_{jl}\right)}{Area\left(\mathcal{V}_j\left(S_i\right)\right)} \\ &= \frac{\max_l Area\left(\mathcal{V}_j\left(S_i\right) \cap M_{jl}\right)}{Area\left(\mathcal{P}_j\left(S_i\right)\right)}. \end{aligned} \tag{4}$$

G. Merging Process

The merging step aims to combine subclusters that constitute the same crop instance. Ideally, for a given view C_j, we would merge a pair of subclusters S_i and $S_{i'}$ if they share the same instance label in M_j. However, we have n such observations from n corresponding views. Therefore, to incorporate all the observations we introduce a subcluster affinity score $\alpha_{ii'}$ between S_i and $S_{i'}$. The score is determined based on the mask reliability scores across n views by

$$\alpha_{ii'} = \sum_{j=1}^{n} r_{ij} r_{i'j} \cdot (-1)^{\mathbb{1}\{\lambda_{ij} \neq \lambda_{i'j}\}}, \tag{5}$$

where $\mathbb{1}\{\cdot\}$ is an indicator function that evaluates to 1 when the condition is satisfied and 0 otherwise. As shown in Fig. 5, for each pair of subclusters in a supercluster we compute the affinity score and build a weighted complete

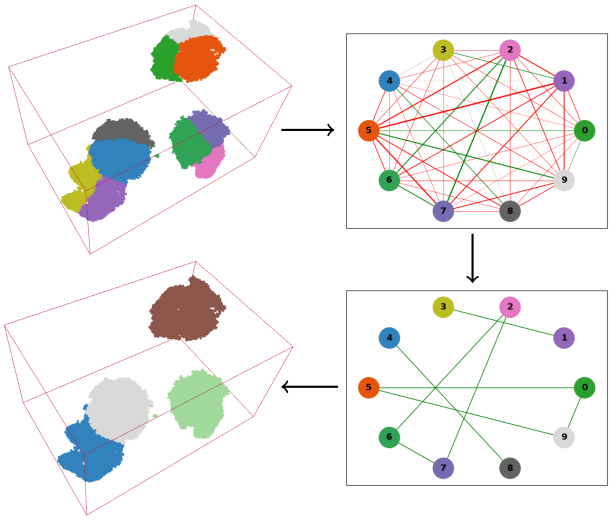

Fig. 5: The subcluster merging process. Top left: a single supercluster is partitioned into 10 subclusters, each represented by a unique color. Top right: a weighted complete graph illustrates the affinities among subclusters, with positive and negative affinities colored in green and red, respectively. The width of each edge is proportional to its affinity value. Bottom right: the graph is partitioned into smaller subgraphs based on affinity scores via label propagation. Bottom left: the corresponding subclusters belonging to the same subgraph are merged.

(a) Cotton plant. (b) Sample views.

Fig. 6: Example images from our cotton plant dataset. (a) shows a cotton plant while (b) displays sample views captured from different viewpoints.

graph. Each subcluster maps to a node in the graph and the affinity score between a pair of subclusters is represented by the corresponding edge weights. Finally, we apply a label propagation algorithm [26] on the graph to partition it into subgraphs that represent the individual crop instances.

IV. Evaluation

A. Cotton Plant Dataset

We collected a cotton plant dataset consisting of 8 plants recorded at the Texas A&M University Research Farm. The images were captured using an Apple iPhone at a resolution of 1040×1920 pixels. Approximately 150 images per plant were taken from a distance of 1 m by recording multiple viewpoints. Fig. 6 depicts representative samples of the camera viewpoints during the data acquisition process. To estimate the image poses, we utilized the Spectacular AI [27] mobile application. The ground-truth counts for each plant were obtained by manually counting all cotton bolls through direct observation. To generate instance masks we employed the Segment Anything Model (SAM) [28], a pretrained 2D instance segmentation model, using manual bounding box supervision as input prompts.

B. Experimental Setup

In addition to evaluating CropNeRF on our cotton plant dataset, we assessed its performance on apple and pear tree data provided by FruitNeRF. The apple tree dataset consists of 3 trees, each photographed using a DSLR camera with a 35 mm lens, producing images at a resolution of 4000×6000 pixels. Roughly 350 images per tree were captured from an approximate distance of 3 m, covering various angles and heights. COLMAP [29] was used to estimate the camera poses. The pear tree dataset was synthetically generated using Blender [30]. The virtual camera was configured with a 35 mm focal length and an image resolution of 1024×1024 pixels. Among the different types of crop data, cotton presents a greater challenge. This is due to the considerable variation in the size and shape of cotton bolls in contrast to the more uniform dimensions of apples and pears. Consequently, our evaluation primarily focuses on cotton.

C. Implementation Details

For the 3D reconstruction and point cloud extraction, we made use of existing NeRF models. Specifically, CropNeRF is based on Nerfacto [31], augmented with a semantic field, and follows the architectural details of FruitNeRF. Given the differences in size among the cotton plants and apple and pear trees, we employed the FruitNeRF network to render the cotton plants while the FruitNeRF-Big network was used to render the apple and pear trees. The models were trained for 30,000 and 100,000 epochs, respectively. To train our model with semantic masks, we derived the masks from the instance masks generated via SAM by assigning a uniform label to all instances. CropNeRF was trained on an Ubuntu 22.04.5 LTS machine with an Intel Xeon Gold 6330 2.00 GHz CPU with 64 GB of memory and an NVIDIA A100 GPU. Training each cotton plant model required about 12 minutes.

To partition the exported point cloud into superclusters, we applied DBSCAN with parameters $eps = 0.02$ and $min\_points = 30$. K-means clustering was performed in the second phase to partition each supercluster into K subclusters where $K = 10$. Notably, the same parameter settings were used for both datasets. In Sec. IV-G, we demonstrate the robustness of CropNeRF by analyzing the relationship between the number of subclusters and the associated counts. To compute the occlusion-aware projection $\mathcal{V}_j(S_i)$ in (1) and (2) we employed a z-buffering technique, which is commonly used in computer graphics. The tree-level count

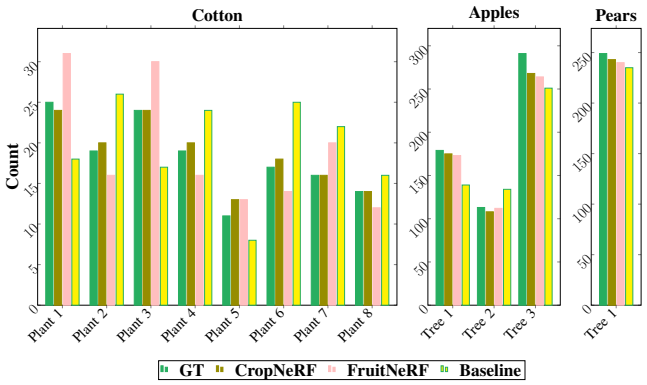

Fig. 7: A cotton boll, apple, and pear counting comparison involving the ground truth (GT), CropNeRF, FruitNeRF, and the baseline.

was obtained by summing the individual instance counts across all superclusters.

D. Counting Accuracy

CropNeRF was tested against a baseline method and recent 3D rendering crop counting approaches using the same input segmentation masks. The baseline technique generates point clouds using COLMAP through an SfM algorithm, followed by segmentation using the clustering method employed in FruitNeRF. Fig. 7 illustrates the per-plant/tree crop counts. For the FruitNeRF results on the apple and pear datasets, we reported the values published in the original study, which used individually tuned per-tree parameter settings. To evaluate FruitNeRF on the cotton dataset, the $template\_size$ parameter was set to 30 mm to be consistent with the average cotton boll diameter reported by Wallace and Fields [32]. We further evaluated CropNeRF using multiple counting error metrics, which are tabulated in Table I.

In comparison to FruitNeRF, CropNeRF consistently demonstrated superior performance. Specifically, CropNeRF achieved a mean absolute percentage error (MAPE) of 4.9% for cotton bolls and 4.7% for apples, indicating a consistent level of accuracy across diverse crop types. On the other hand, FruitNeRF yielded a MAPE of 18.1% for cotton bolls and 4.5% for apples, revealing a significant variation in performance between the two crops. This deviation stems from the dependency of clustering-based point cloud segmentation methods on predefined template crop sizes and shapes. For pears, both frameworks achieved relatively low MAPE values (2.4% for CropNeRF versus 3.6% for FruitNeRF) compared to apples and cotton bolls, which can be attributed to the fact that pears typically grow individually rather than in dense clusters. Compared to Cotton3DGaussians [22], CropNeRF produced lower counting errors despite operating in a real-world outdoor environment. These results highlight the robustness and versatility of CropNeRF for counting tasks across different agricultural commodities.

E. Mask Discrepancy Handling

As displayed in Fig. 8a, instance masks generated by SAM can exhibit inconsistencies, i.e., they do not fully align with the corresponding instances. Additionally, multiple distinct

Method	Cotton Bolls		Apples		Pears	
	R	**M**	**R**	**M**	**R**	**M**
Baseline	5.9	30.8	34.8	18.2	14.0	5.6
FruitNeRF [21]	3.9	18.1	15.9	**4.5**	9	3.6
Cotton3DGaussians [22]	1.7	9.2	-	-	-	-
CropNeRF (ours)	**1.0**	**4.9**	**13.7**	4.7	**6.0**	**2.4**

TABLE I: A comparison of counting errors across different methods. **R** stands for root mean squared error and **M** signifies mean absolute percentage error.

Fig. 8: Examples of mask inconsistencies where two cotton boll instances are inconsistently labeled across multiple views: (a) the mask does not align with the corresponding instance boundary; (b-e) two bolls are incorrectly labeled as a single instance; (f) one of the bolls is missed by the detector; (g) bolls are correctly labeled; (h) CropNeRF's reliability score attenuates inconsistencies and allows for successful identification of the bolls.

instances may be erroneously labeled as a single instance (e.g., Fig. 8b-8e), while in other cases certain instances may remain unlabeled due to misdetection (e.g., Fig. 8f). These discrepancies are especially evident in the cotton plant dataset, where the irregular shape of the bolls makes it challenging to differentiate individual instances precisely.

CropNeRF is able to mitigate many of these errors and successfully identify crop instances, Fig. 8h. For instance, the reliability score is low in scenarios such as Fig. 8a and Fig. 8f due to low mask consistency. Similarly, scenarios such as Fig. 8c, Fig. 8d, and Fig. 8e yield lower reliability scores due to reduced visibility scores. With lower reliability scores these scenarios have a diminishing effect on subsequent affinity calculations, resulting in segmentations that are robust to mask discrepancies.

F. Instance Mask Effectiveness

We evaluated the effectiveness of CropNeRF on counting tasks using a sparse set of instance masks. In this experiment, we varied the number of available masks m from 5 to 150 and analyzed the resulting boll counting error across different cotton plants. For each value of m, a set of m masks is uniformly sampled from the complete set of 150 masks. The results presented in Fig. 9 indicate that in most cases only 15 to 30 masks are sufficient to achieve an accurate count.

G. Number of Subclusters Impact

A primary goal in designing CropNeRF was to minimize the need for crop-specific parameter optimization. Accord-

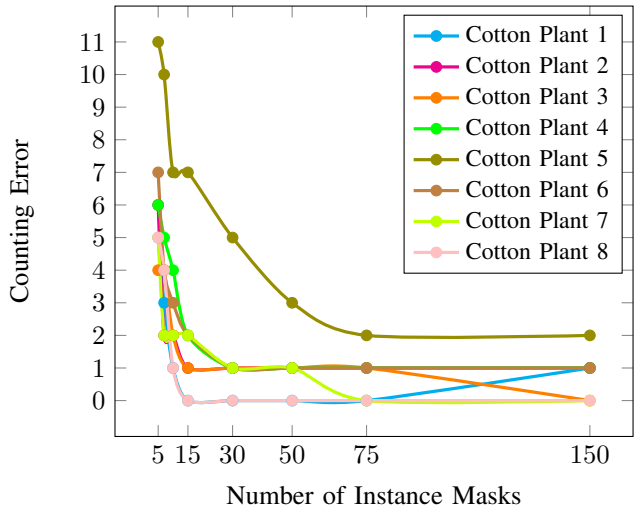

Fig. 9: The number of instance masks versus the counting error.

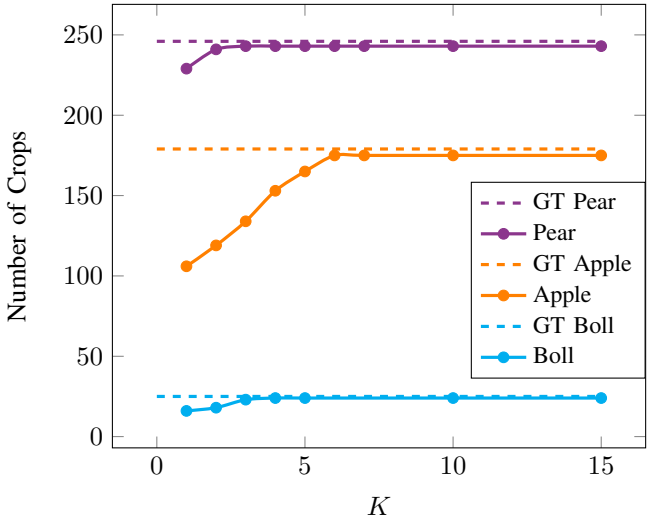

Fig. 10: The number of subclusters (K) versus the crop count. The dashed line represents the ground-truth (GT) count while the solid line indicates the number of crops identified by CropNeRF.

ingly, we analyzed the relationship between the parameter K, a key input to the k-means clustering algorithm, and the counting accuracy. Samples from the apple tree, pear tree, and cotton plant datasets were selected to evaluate the influence of K. The results are plotted in Fig. 10.

For cotton bolls, the results show that the count remained consistent for $K \geq 4$. This stability arises from the fact that in most cases the number of instances within a supercluster does not exceed five. The maximum number of bolls observed within a single supercluster was four for the cotton plant dataset. Similarly, for apples the count stayed approximately constant with $K \geq 6$ as the apple clusters typically contained a maximum of six to seven instances. In the case of pears, clusters generally comprised only one or two pears, leading to stable counts for $K \geq 2$.

Based on these empirical findings, we set $K = 10$ for all the experiments. In general, we recommend selecting K to be greater than the expected number of instances within a cluster. Nevertheless, K should not be excessively large since this will increase computational complexity due to the higher number of subclusters.

Method	Visibility	Mask	LPA	MAPE% ↓
Baseline				7.1
+visibility	✓			6.3
+mask		✓		6.5
+mask,+visibility	✓	✓		5.4
CropNeRF	✓	✓	✓	**4.9**

TABLE II: An ablation study on the cotton plant dataset. The first row is the baseline method, which merges subclusters using only the instance mask. In subsequent rows, the integration of visibility and mask consistency scores progressively enhances the counting accuracy. The last row, representing CropNeRF, applies a label propagation algorithm (LPA) in addition to visibility and mask consistency scores resulting in the lowest error.

H. Ablation Study

To understand the impact of key design choices in CropNeRF, we examined the following components: visibility score, mask consistency score, and label propagation algorithm. The ablation study results are presented in Table II and described as follows. The baseline method omits the visibility and mask consistency scores. Instead, it merges subcluster pairs if and only if $\alpha_{ii'} > 0$, where $\alpha_{ii'} = \sum_{j=1}^{n}(-1)^{\mathbb{1}\{\lambda_{ij} \neq \lambda_{i'j}\}}$. The +visible and +mask methods, shown in the second and third rows respectively, compute an affinity score based on either the visibility score $\alpha_{ii'} = \sum_{j=1}^{n} v_{ij} \cdot (-1)^{\mathbb{1}\{\lambda_{ij} \neq \lambda_{i'j}\}}$ or the mask consistency score $\alpha_{ii'} = \sum_{j=1}^{n} c_{ij} \cdot (-1)^{\mathbb{1}\{\lambda_{ij} \neq \lambda_{i'j}\}}$. Subcluster pairs are merged if and only if $\alpha_{ii'} > 0$.

CropNeRF incorporates both visibility and mask consistency scores to compute a reliability score along with a label propagation strategy for subcluster merging. The ablation study results highlight that the visibility score is the dominant factor in determining segmentation performance followed by the mask consistency score. Consequently, when affinity scores are derived from these two factors, merging if and only if $\alpha_{ii'} > 0$ is sufficient for achieving a precise segmentation.

V. Limitations and Future Work

Although NeRFs and their variants can produce high-fidelity 3D representations, they require substantial computing resources. In the CropNeRF pipeline, NeRF model training is the most computationally intensive and time-consuming stage. Furthermore, accurately detecting certain crops within dense canopy structures remains difficult due to the spectral limitations of RGB cameras. Future work will not only focus on substitutes to NeRFs, but also on integrating additional sensing modalities such as normalized difference vegetation index and thermal imaging to enhance crop detection and improve the overall robustness of the framework.

VI. Conclusion

This paper introduced CropNeRF, an occlusion-aware counting framework that accurately enumerates individual crop instances via 3D instance segmentation. Furthermore, we created a multi-view cotton plant image dataset to promote future research in this area. CropNeRF is highly robust and does not require sensitive parameter tuning for different crop types. We evaluated the effectiveness of CropNeRF on three distinct crop datasets (cotton bolls, apples, and pears), which drastically vary in color, shape, and size. Experimental results demonstrate accurate counts that exceed the state of the art across all three datasets, highlighting framework's adaptability and reliability for agricultural applications.

Acknowledgments

Md Ahmed Al Muzaddid was supported by a University of Texas at Arlington Dissertation Fellowship. We thank Aaron J. DeSalvio for assistance with the data collection at Texas A&M University. We acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing software, computational, and storage resources that have contributed to these results.

References

[1] L. He, W. Fang, G. Zhao, Z. Wu, L. Fu, R. Li, Y. Majeed, and J. Dhupia, "Fruit yield prediction and estimation in orchards: A state-of-the-art comprehensive review for both direct and indirect methods," *Computers and Electronics in Agriculture*, vol. 195, p. 106812, 2022.

[2] Y. Liu, B. Hu, J. Huang, Y.-W. Tai, and C.-K. Tang, "Instance neural radiance field," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 787–796.

[3] M. Ye, M. Danelljan, F. Yu, and L. Ke, "Gaussian grouping: Segment and edit anything in 3d scenes," in *Proceedings of the European Conference Computer Vision*. Springer, 2024, pp. 162–179.

[4] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor, and V. Kumar, "Counting apples and oranges with deep learning: A data-driven approach," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 781–788, 2017.

[5] N. Häni, P. Roy, and V. Isler, "Apple counting using convolutional neural networks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 2559–2565.

[6] D. Tedesco-Oliveira, R. P. da Silva, W. Maldonado Jr, and C. Zerbato, "Convolutional neural networks in predicting cotton yield from images of commercial fields," *Computers and Electronics in Agriculture*, vol. 171, p. 105307, 2020.

[7] 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.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference Computer Vision*. Springer, 2016, pp. 21–37.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[10] J. A. James, H. K. Manching, M. R. Mattia, K. D. Bowman, A. M. Hulse-Kemp, and W. J. Beksi, "Citdet: A benchmark dataset for citrus fruit detection," *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 10 788–10 795, 2024.

[11] J. A. James, H. K. Manching, A. M. Hulse-Kemp, and W. J. Beksi, "Few-shot fruit segmentation via transfer learning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2024, pp. 13 618–13 624.

[12] C. Smitt, M. Halstead, T. Zaenker, M. Bennewitz, and C. McCool, "Pathobot: A robot for glasshouse crop phenotyping and intervention," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2021, pp. 2324–2330.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017, pp. 2961–2969.

[14] W. Zhang, J. Wang, Y. Liu, K. Chen, H. Li, Y. Duan, W. Wu, Y. Shi, and W. Guo, "Deep-learning-based in-field citrus fruit detection and tracking," *Horticulture Research*, vol. 9, p. uhac003, 2022.

[15] J. Redmon, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[16] M. A. Al Muzaddid and W. J. Beksi, "Ntrack: A multiple-object tracker and dataset for infield cotton boll counting," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 9, pp. 7452–7464, 2024.

[17] M. A. Al Muzaddid, J. A. James, and W. J. Beksi, "Croptrack: A tracking with re-identification framework for precision agriculture," *arXiv preprint arXiv:2512.24838*, 2025.

[18] S. Sun, C. Li, P. W. Chee, A. H. Paterson, Y. Jiang, R. Xu, J. S. Robertson, J. Adhikari, and T. Shehzad, "Three-dimensional photogrammetric mapping of cotton bolls in situ based on point cloud segmentation and clustering," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 160, pp. 195–207, 2020.

[19] G. P. Matos, C. Santiago, J. P. Costeira, R. L. Saldanha, and E. M. Morgado, "Tracking and counting apples in orchards under intermittent occlusions and low frame rates," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5413–5421.

[20] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[21] L. Meyer, A. Gilson, U. Schmid, and M. Stamminger, "Fruitnerf: A unified neural radiance field based fruit counting framework," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 1–8.

[22] L. Jiang, J. Sun, P. W. Chee, C. Li, and L. Fu, "Cotton3dgaussians: Multiview 3d gaussian splatting for boll mapping and plant architecture analysis," *Computers and Electronics in Agriculture*, vol. 234, p. 110293, 2025.

[23] D. Zhang, J. Gajardo, T. Medic, I. Katircioglu, M. Boss, N. Kirchgessner, A. Walter, and L. Roth, "Wheat3dgs: In-field 3d reconstruction, instance segmentation and phenotyping of wheat heads with gaussian splatting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 5360–5370.

[24] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, "In-place scene labelling and understanding with implicit scene representation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 838–15 847.

[25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, vol. 96, no. 34, 1996, pp. 226–231.

[26] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, no. 3, p. 036106, 2007.

[27] (2026) Spectacular AI. [Online]. Available: https://github.com/spectacularai

[28] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[29] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.

[30] (2026) Blender. [Online]. Available: https://www.blender.org

[31] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, J. Kerr, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development," in *Proceedings of the ACM SIGGRAPH Conference*, 2023, pp. 1–12.

[32] B. W. Wallace and C. A. Fields, "Boll diameter and seed number relation to seedcotton weight per boll," in *Proceedings of the Beltwide Cotton Conference*, vol. 2, 1997, pp. 1444–1446.