

Plant Taxonomy Meets Plant Counting: A Fine-Grained, Taxonomic Dataset for Counting Hundreds of Plant Species

Supplementary Material

The supplementary material provides the algorithms for dataset partitioning, the method for mapping the hierarchical taxonomic structure, qualitative visualizations and some image examples. First, **Sec. A** details the Mixed-Integer Linear Programming (MILP) formulation for dataset partitioning. Second, **Sec. B** explains how we construct the mapping from taxonomic names to hierarchical codes. Finally, **Sec. C** presents both qualitative comparisons with baselines and diverse visual examples.

A. MILP-based Dataset Partitioning

We propose a Mixed-Integer Linear Programming (MILP) approach to partition the dataset. This ensures taxonomic independence between subsets while maintaining statistical balance in instance density.

A.1. Process Description

The pipeline consists of three stages:

Build Units. We group images into atomic units based on unique *Species-Organization* pairs. For example, all images of *Zea mays-flower* form a single unit. Each atomic unit corresponds to a unique visual phenotype of a single species and is assigned exclusively to one subset.

Optimize Assignments. The unit assignments are determined using the PULP_CBC_CMD solver. The optimization objective combines two types of deviation penalties. First, we minimize how far the image counts (N_k) deviate from the target 7 : 1 : 2 proportion, ensuring that the group sizes remain close to this desired ratio. Second, we minimize the imbalance in the average number of object instances (points) (P_k) across groups. Since matching the image ratio is more important, we assign it a much larger weight ($\lambda_{img} = 100.0$) compared to the weight for balancing instance counts ($\lambda_{pts} = 1.0$). This forces the solver to satisfy the image ratio first, and only then adjust the instance distribution.

Ensure Coverage. To ensure that the model is exposed to all levels of biological organization, we enforce a hard constraint that the training split must include at least one unit from every category. After the solver completes, we also verify that each of the three subsets contains data from all observation scales (Microscopy, Close-range, Remote Sensing). If any subset lacks a particular scale, the partition is rejected and the optimization is restarted with a different random seed for the solver’s internal heuristics, ensuring a new search trajectory and a fresh candidate solution.

A.2. Problem Formulation

Let \mathcal{D} be the full dataset containing images I , and let $\mathcal{U} = \{u_1, \dots, u_M\}$ denote the set of atomic units, where each unit corresponds to a unique *Species-Organization* pair. For each unit u , we denote by n_u the number of images and by p_u the total number of object instances.

Let \mathcal{O} be the set of biological organization categories, and let $\mathcal{U}_o \subseteq \mathcal{U}$ denote the units belonging to category o . Similarly, let \mathcal{T} be the set of observation scales (Microscopy, Close-range, Remote Sensing), and let $\mathcal{U}_t \subseteq \mathcal{U}$ denote the units originating from scale t .

We partition \mathcal{U} into three subsets $\mathcal{S} = \{\text{train, val, test}\}$ with target proportions $r_k = \{0.7, 0.1, 0.2\}$.

Decision Variables. For each unit u and split k , we define a binary assignment variable:

$$x_{u,k} = \begin{cases} 1, & \text{if unit } u \text{ is assigned to split } k, \\ 0, & \text{otherwise.} \end{cases}$$

Objective Function. We minimize the weighted deviation between the actual and target totals for image counts (\hat{N}_k) and point counts (\hat{P}_k). The objective function is formulated as:

$$\min \sum_{k \in \mathcal{S}} \left(\lambda_{img} \left| \sum_{u \in \mathcal{U}} n_u x_{u,k} - \hat{N}_k \right| + \lambda_{pts} \left| \sum_{u \in \mathcal{U}} p_u x_{u,k} - \hat{P}_k \right| \right) \quad (1)$$

where $\hat{N}_k = r_k \sum_u n_u$ and $\hat{P}_k = r_k \sum_u p_u$. We use $\lambda_{img} = 100$ and $\lambda_{pts} = 1$ to strictly prioritize the adherence to the image-count ratio.

Constraints. The optimization is subject to the following constraints:

- i) *Unique assignment.* Each unit must be assigned to exactly one split:

$$\sum_{k \in \mathcal{S}} x_{u,k} = 1, \quad \forall u \in \mathcal{U}. \quad (2)$$

- ii) *Training set coverage.* To ensure robust feature learning, the training subset is explicitly constrained to in-

clude at least one unit from each organization category:

$$\sum_{u \in \mathcal{U}_o} x_{u, \text{train}} \geq 1, \quad \forall o \in \mathcal{O}. \quad (3)$$

Verification. While the MILP guarantees organization coverage for the training set, we additionally require that all splits cover the full spectrum of observation scales. After optimization, we verify that:

$$\sum_{u \in \mathcal{U}_t} x_{u, k} \geq 1, \quad \forall t \in \mathcal{T}, \forall k \in \mathcal{S}. \quad (4)$$

If this condition is not met for any split k , the resulting partition is discarded, and the optimization is re-initialized with a different random seed until a valid solution is obtained.

B. Hierarchical Taxonomic Encoding

We follow the standard 7-hierarchy biological taxonomy—*Kingdom*, *Phylum*, *Class*, *Order*, *Family*, *Genus*, and *Species*. To build the index mappings, we first sort all species names in the dataset and iterate through them in order. For each species, we retrieve its taxonomic labels across all remaining hierarchies. Whenever a hierarchy encounters a category name for the first time, we append it to that level’s dictionary and assign the next integer index starting from 1. Formally, for each hierarchy h , this procedure defines a mapping

$$\phi_h : \mathcal{C}_h \rightarrow \{1, 2, \dots, |\mathcal{C}_h|\},$$

where \mathcal{C}_h denotes the set of unique category names observed at hierarchy h . This construction yields a complete mapping from taxonomic names to integer identifiers for every hierarchy.

Under this protocol, each species is encoded as a 7-dimensional vector v :

$$v = [id_{\text{king}}, id_{\text{phy}}, id_{\text{cls}}, id_{\text{ord}}, id_{\text{fam}}, id_{\text{gen}}, id_{\text{spec}}]$$

where each component id indicates the integer index of the taxon at that specific hierarchy.

Example. Consider the species *Malus domestica*. Its taxonomic path maps to the following indices:

Kingdom: *Plantae* \rightarrow 1; Phylum: *Tracheophyta* \rightarrow 1; Class: *Magnoliopsida* \rightarrow 1; Order: *Rosales* \rightarrow 14; Family: *Rosaceae* \rightarrow 39; Genus: *Malus* \rightarrow 113; Species: *Malus domestica* \rightarrow 136.

Consequently, the hierarchical encoding vector is:

$$v_{\text{Malus domestica}} = [1, 1, 1, 14, 39, 113, 136]$$

The file `taxonomy_ids.json` contains the complete index mappings for all categories across all 7 taxonomic

hierarchies. This file is included with the supplementary materials, and its structure is illustrated below:

```
{
  "Kingdom": {
    "Plantae": 1,
    "Fungi": 2
  },
  "Phylum": {
    "Tracheophyta": 1,
    "Basidiomycota": 2
  },
  ...
  "Family": {
    "Malvaceae": 1,
    "Fabaceae": 2,
    ...
    "Rhamnaceae": 83
  },
  ...
  "Species": {
    "Abelmoschus esculentus": 1,
    ...
    "Zea mays": 240,
    "Ziziphus mauritiana": 242
  }
}
```

C. Visualizations and Examples

Fig. 7 shows images of a single species observed at different growth stages, where the changes in plant size, structure, and overall appearance become clearly visible across the timeline.

Fig. 8 presents examples from several biological organizations of the same species. The differences in texture, geometry, and visual scale across tissues, organs, and whole plants are evident from these samples.

Fig. 9 includes images collected under low-light conditions. Reduced visibility and unstable illumination make object boundaries harder to distinguish.

Fig. 10 shows real-world scenes, with irregular backgrounds, occlusions, and surrounding environment adding substantial visual clutter.

Fig. 11 provides representative high-density cases containing large numbers of closely arranged instances. The crowded layouts and overlapping structures make accurate separation and counting particularly difficult.

As shown in Fig. 12, we include qualitative comparisons across multiple baseline methods to illustrate their performance under different plant-counting conditions.



Figure 7. **Examples of the same species at different growth stages in our TPC-268.** The images span early seedling, vegetative development, tasseling, and final maturity.



Figure 8. **Examples of different biological organizations from the same species in our TPC-268.** Tissue-level, organ-level, and whole-plant-level images show distinct texture patterns and structural characteristics.



Figure 9. **Examples of images captured in dark or low-illumination environments from our TPC-268.** Reduced contrast, color distortion, and shadow-induced ambiguity complicate instance perception.



Figure 10. **Examples of real-world scenarios in our TPC-268.** Natural backgrounds, occluding structures, and cluttered surroundings reflect practical field conditions.



Figure 11. **Examples of high-density counting scenarios in our TPC-268.** These images include scenes containing hundreds or even thousands of instances, characterized by compact spatial arrangement, heavy overlap, and minimal inter-instance separation.

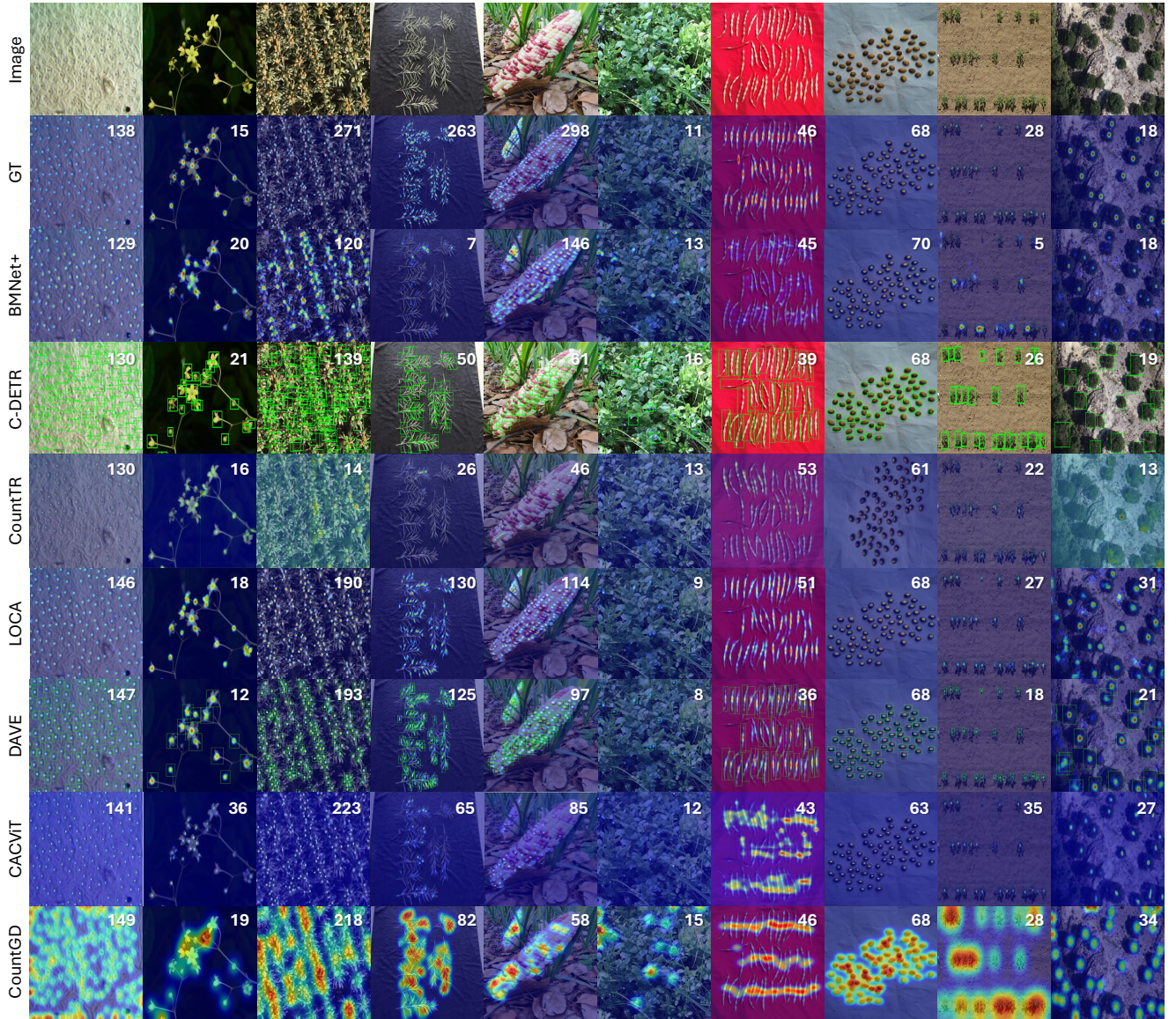


Figure 12. **Qualitative results of representative counting methods on our TPC-268.** The examples cover diverse plant forms, observation scales, and density levels.