A Unified Transformer-based Framework for Panoptic Segmentation of Crops and Weeds - VIDAR -

Qi Chen[1] Jingjing Fu[2] Zhiwei Xiong[1]
[1]University of Science and Technology of China
[2]Microsoft Research Asia

1. Team details

- Team name
  VIDAR
- Team leader name
  Qi Chen
- Team leader address, phone number, and email
  - Address: Dept. EEIS, P.O.Box 4, Hefei 230027, Anhui, China
  - Phone: (86) 15869825609
  - qic@mail.ustc.edu.cn
- Team members
  Qi Chen, Jingjing Fu, Zhiwei Xiong
- Team website URL (if any)
  http://vidar-ustc.github.io
- Affiliation
  University of Science and Technology of China
- User names and entries on the CVPPA@ICCV'23: Hierarchical Panoptic Segmentation of Crops and Weeds Codalab competitions
  qicq1c
- Best scoring entries of the team during test phase
  PQ+ 78.30, PQ 71.72
- Link to the codes/executables of the solution(s)
  To be added

2. Contribution details

- Title of the contribution
  A Unified Transformer-based Framework for Panoptic Segmentation of Crops and Weeds
- General method description
  We divide the hierarchical panoptic segmentation challenge into two distinct subtasks: plant panoptic segmentation and leaf panoptic segmentation. The plant panoptic segmentation necessitates generating semantic masks that allocate each pixel to crop, weed, or soil, in addition to instance segmentation for crops and weeds. The leaf panoptic segmentation entails producing semantic masks that assign each pixel to leaf or soil, coupled with instance segmentation for leaves. We implement a query-based panoptic segmentation network, predominantly inspired by [1]. Our strategy employs queries to cooperatively predict bounding boxes and masks for all classes in both panoptic segmentation tasks, leading to more discriminative features.
- Representative image / diagram of the method(s)

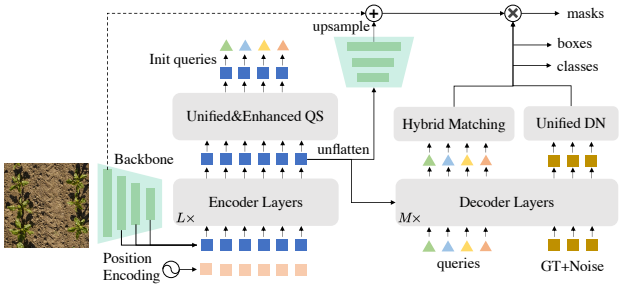

Figure 1. The framework of our approach. Our approach is completely based on the framework of MaskDINO, and the specific network details can be referred to [1].

3. Global Method Description

- Which pretrained or external methods/models have been used (for any stage, if any)
  No pre-trained or external methods / models.
- Which additional data has been used in addition to the provided training and validation data (at any stage, if

any)
No additional data is used.

- Training description
  - We use ResNet-50 as the backbone and follow the same multi-scale setting as in MaskDINO [1], using 4 scales of ResNet-50 for the mask decoder. For data augmentation, we apply large-scale jittering (LSJ) and flip augmentations, along with a fixed-size crop of 1024×1024. The batch size is set to 8, and we use the ADAMW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. A weight decay of 0.05 is applied to all layers. The WarmupPoly learning rate schedule is adopted, featuring an initial learning rate of 0.0001 and linear warming up during the first 1000 iterations.
  - We jointly train detection and segmentation tasks, incorporating a total of three types of losses: classification loss $\mathcal{L}_{cls}$, box loss $\mathcal{L}_{box}$, and mask loss $\mathcal{L}_{mask}$. The box loss, which consists of L1 loss and GIOU loss, and the classification loss, which uses focal loss, are consistent with the losses used in MaskDINO [1]. For mask loss, we employ cross-entropy $\mathcal{L}_{ce}$ and dice loss $\mathcal{L}_{dice}$. Additionally, we incorporate point loss in mask loss to improve efficiency. The total loss is a linear combination of the three types of losses: $\lambda_{cls}\mathcal{L}_{cls} + \lambda_{L1}\mathcal{L}_{L1} + \lambda_{giou}\mathcal{L}_{giou} + \lambda_{ce}\mathcal{L}_{ce} + \lambda_{dice}\mathcal{L}_{dice}$. We set the loss weights as follows: $\lambda_{cls} = 4, \lambda_{giou} = 2, \lambda_{ce} = \lambda_{dice} = \lambda_{L1} = 5$.
  - For the plant panoptic segmentation task, we configure the query number to be 100. Meanwhile, for the leaf panoptic segmentation task, we set the query number to 300. We carry out a total of 25k iterations for both tasks.
  - We implement the proposed networks with pytorch framework and train them using four 32G V100 GPUs.
- Testing description
  During the inference phase, we utilize test time augmentation for semantic segmentation of the stuff class. Furthermore, we apply the instance segmentation inference approach from MaskDINO to predict the masks for the thing class.

4. Technical details

- Language and implementation details (including platform, memory, parallelization requirements)
  - Platform: Python3.6+, Pytorch 1.9+
  - 32G memory, Tesla V100
- Human effort required for implementation, training and validation?
  None.
- Training/testing time? Runtime at test per image.
  - Training time: 12 hours on 4 32G Tesla V100 for training 25k iterations.
  - Test time: 0.5s per image on Tesla V100.
- The efficiency of the proposed solution(s).
  This is not a very high-efficient solution.

5. Other details

- General comments and impressions of the CVPPA@ICCV'23: Hierarchical Panoptic Segmentation of Crops and Weeds challenge.
  We thank the organizers for their hard work.

References

[1] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3041–3050, 2023. 1, 2