MinDVPS: Minimal Model for Depth-aware Video Panoptic Segmentation

Kim Ji-Yeon[2] Oh Hyun-Bin[1] Dahun Kim[5] Tae-Hyun Oh[1,2,3,4]

[1]Dept. of Electrical Engineering, [2]Convergence IT Engineering and
[3]Grad. School of Artificial Intelligence, POSTECH, Korea
[4]Institute for Convergence Research and Education in Advanced Tech., Yonsei University, Korea
[5]Google DeepMind

Abstract

Depth-aware Video Panoptic Segmentation (DVPS) is one of the complicated multi-task learning problems that jointly tackles video panoptic segmentation and depth estimation in a single model. Existing works are typically composed of task-specialized heads, including respective segmentation heads for things and stuff, global and instance depth map heads, and a tracking head, and are trained with consecutive video frames. Increasing the complexity of modules, losses, and data batch may lead to sensitive performance against training or hyper-parameter configurations. In this work, we attempt to seek for minimal architecture and configurations for the DVPS task. Motivated by the past success of the per-frame semantic segmentation methods in the video semantic segmentation field, we propose MinDVPS, a simple and minimal model that does not require any temporal annotations during training and the tracking module. Instead of using extra tracking modules, our model utilizes the learnable embeddings, i.e., queries, to track the objects frame-by-frame in online fashion. We also demonstrate the effectiveness of our design choice by achieving the state-of-the-art performances on Cityscapes-DVPS.

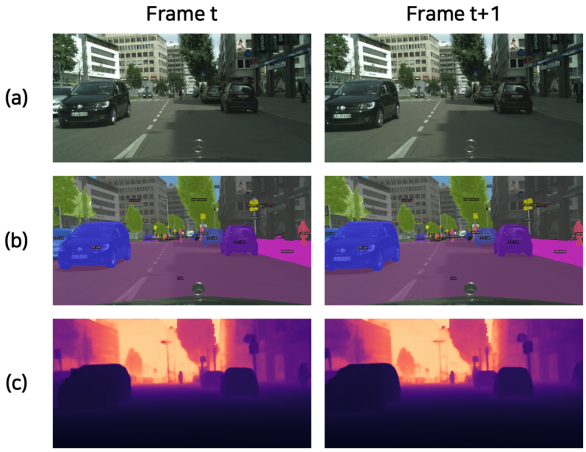

Figure 1. **Depth-aware Video Panoptic Segmentation**. (a) Input image samples. (b) Panoptic segmentation results with semantic classes and instance ID and (c) monocular depth estimation results obtained by our method. Despite training over independent images not consecutive video frames, our MinDVPS predicts both panoptic segmentation and monocular depth maps in a temporally consistent manner over consecutive frames during inference.

1. Introduction

As humans can recognize and process multiple information at once, building a model for multi-task learning is essential in computer vision. While the single-task models, *e.g.* [6, 9, 15, 20], are diverged into their own domains, the multi-task models, *e.g.* [19, 22], explore broader perspectives and exploit mutual information from different tasks. In particular, in autonomous driving scenarios, the multi-task ability is indispensable to facilitate a holistic scene understanding, *e.g.*, detecting, segmenting, tracking objects, and computing the distance of the objects from the camera; *i.e.*, Depth-aware Video Panoptic Segmentation (DVPS).

DVPS tackles both video panoptic segmentation and depth estimation, mostly in driving scenes, as shown in Fig. 1. DVPS aims to assign a class, ID, and depth value to each pixel consistently along the consecutive video frames. DVPS is inherently challenging due to its multi-task learning form, where it requires to satisfy independent criteria of each task that might conflict with each other, *e.g.*, to accurately segment objects, precisely estimate depth, and correctly track objects. It should also consider temporal consistency across the frames.

Recent works [14, 23] have been conducted to resolve these challenges by adopting module-wise architecture and adding extra tracking modules to deal with consecutive input frames. Particularly, PolyphonicFormer [23] utilizes an additional tracking loss to train the extra tracking modules. It introduces careful loss balancing and a recipe for training

dataset sequence during training; otherwise, it might fail to exploit mutual information and disturb finding the optimal point of multi-task. Moreover, adopting additional tracking modules causes extra computational costs, which may limit the applicability.

In this work, we propose MinDVPS, a minimal model for depth-aware video panoptic segmentation. Our work is motivated by MinVIS [8], where they argue video-based architecture and training are not required for video instance segmentation performances. They show that only training on the per-frame instance segmentation task can surpass video-based architectures. We apply the same finding to DVPS, propose an end-to-end per-frame based learning architecture for DVPS, and see whether the same observation holds. Without adopting extra tracking modules, our model is trained with frame-level annotations just as an image-based network, *i.e*., per-frame learning. Our model tracks the objects across the frame by linking the learnable embeddings, *i.e*. queries. We conduct experiments with our model, MinDVPS, on Cityscapes-DVPS [14] to see the effectiveness of our design choice, and achieve the state-of-the-art performance. Thereby, we show that the same observation with MinVIS [8] is indeed shared even in our DVPS setup which is more complex than their task.

2. Related Work

Per-frame learning for video segmentation. Recent video segmentation techniques can be categorized into two video processing approaches: *per-clip* and *per-frame*. The per-clip based works [3, 11, 16, 18] involve directly processing the entire spatial-temporal volume of a video to predict spatial-temporal masks for each object instance. Although these per-clip methods have led to significant improvements in video segmentation, they require substantial memory and computational resources to deal with longer video clips.

On the other hand, the per-frame based works [8, 17, 20, 21] segment object instances for each frame and then temporally match them using post-processing steps, often accompanied by heuristics like tracking modules. Woo *et al*. [17] process video frames at inference time with a per-frame model by training with two different levels of temporal correspondence objective, *i.e*., segment and pixel. MinVIS [8] does not have any temporal cues when training the image instance segmentation model and matches corresponding queries of each frame during inference without the need for manually designed tracking modules. They achieve comparable or even outperforming results without requiring annotations for all frames in a video. Given these advantages, we adopt the per-frame approach of MinVIS and extend it to the multi-task model for predicting both panoptic segmentation and monocular depth estimation.

Depth-aware video panoptic segmentation. Many recent approaches have been studied for joint learning of video panoptic segmentation and depth estimation. ViP-DeepLab [14] introduces the DVPS task and releases the datasets by adding depth annotations to Cityscapes-VPS [9] and SemanticKITTI [1]. MonoDVPS [13] suggests a method for self-supervised monocular depth estimation in DVPS and introduce segmentation-guided depth loss to improve depth prediction. PolyphonicFormer [23] unifies video panoptic segmentation and depth estimation tasks using a vision transformer and query-based learning. They demonstrate the effectiveness of DETR [2]-like architecture in DVPS, exhibiting remarkable performance. We also adopt query-based transformer architecture while we do not need any tracking module and temporal annotations in training, which is simpler and more effective.

3. Method

3.1. Overview

Our model takes a single frame as an input and outputs both the segmented map and the estimated depth map. Given the input image $I \in \mathbb{R}^{H \times W}$, our model aims to predict a set of N segments as $set\ \mathcal{Y}_i = \{(m_i, p_i(c), d_i)\}_{i=1}^{N}$, where $m_i \in [0,1]^{H \times W}$ denotes the mask of the segment i, $p_i(c)$ denotes the probability of the segment assigned to the class category $c \in \{1, \ldots, K+1\}$ and $d_i \in [0, d_{max}]^{H \times W}$ denotes the depth map of the segment. The $K+1$ class categories contain a "no-object" category ($\varnothing$) for the segment not corresponding to any region and K ground-truth classes. We set $N = 100$ to be large enough to cover all the things and stuff instances appearing in the frame ($N \gg K$).

3.2. Per-frame learning of MinDVPS

Our model does not require any temporal annotations while training. Instead of adopting extra tracking modules to handle the temporal cues, we only use frame-level cues to guide the model to learn the representations of the objects in the frame. If the model is well-trained enough to distinguish the representations in the frame and to be consistent across the adjacent frames, we can use these representations for object tracking [8].

Motivated by DETR [2], we utilize the queries to learn the object representations. Each branch takes the initial N queries respectively and encodes the task-specific information into the queries. For the panoptic segmentation, each query learns the class and the mask information of each object. We use bipartite matching to assign N predictions to the ground-truth instances with the matching cost as in previous work [8]. After matching, we use two loss terms for the segmentation branch as

$$\mathcal{L}_{seg} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{mask}\mathcal{L}_{mask}. \quad (1)$$

We use cross-entropy loss for $\mathcal{L}_{cls}$, and the summation of the

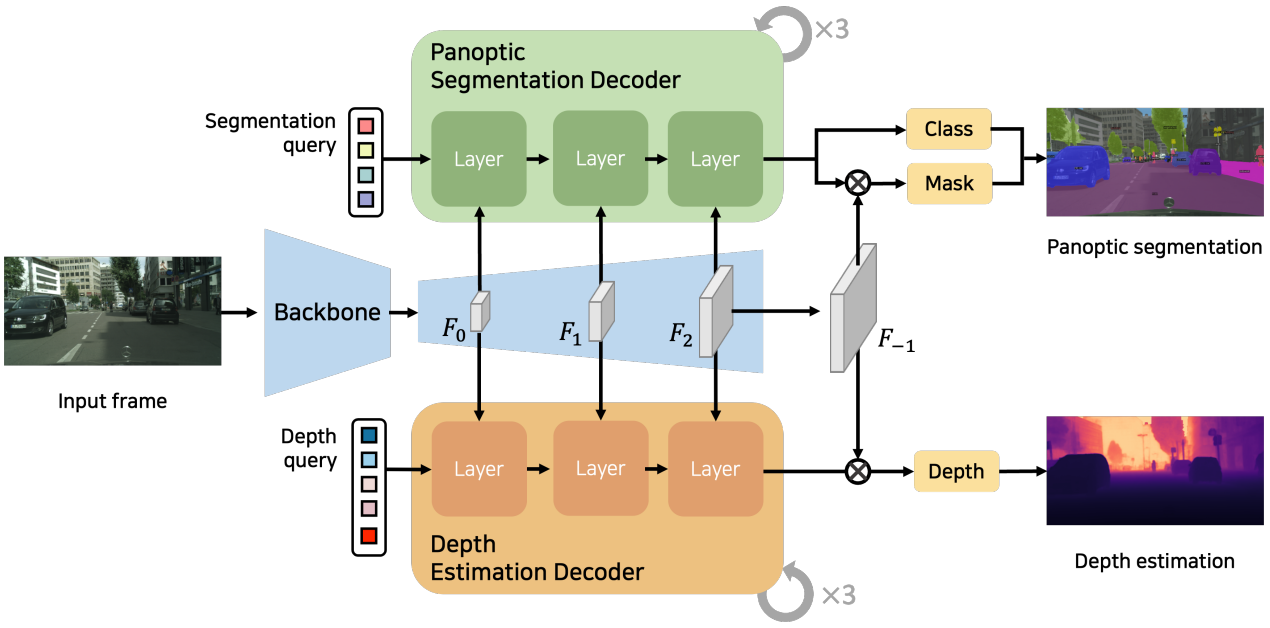

Figure 2. **Overall architecture**. Given input frame, our image encoder extracts the multi-scale image features from the input. Then the feature and task-speific queries passed into the segmentation/depth decoder. In the decoder, the queries are updated and decoded into final panoptic segmentation map and depth map, respectively.

binary cross-entropy loss and the dice loss [12] for $\mathcal{L}_{mask}$. For the depth estimation, each query learns the relative depth of each instance. We apply the same matching results from the segmentation to the depth estimation so that depth queries have the same correspondence as the segmentation queries. The depth loss is composed of scale-invariant logarithmic loss [4], absolute relative loss, and square relative loss [5] as

$$\mathcal{L}_{depth} = \lambda_{si}\mathcal{L}_{si} + \lambda_{abs}\mathcal{L}_{abs} + \lambda_{sq}\mathcal{L}_{sq}. \quad (2)$$

The total loss is composed of the segmentation and the depth losses as

$$\mathcal{L}_{total} = \lambda_{seg}\mathcal{L}_{seg} + \lambda_{depth}\mathcal{L}_{depth}. \quad (3)$$

Here, $\{\lambda_*\}$ denotes the weight parameters for each loss term.

3.3. Architecture

MinDVPS is a minimal query-based transformer architecture for depth-aware video panoptic segmentation. The overall architecture is described in Fig. 2. Our model contains three main modules: image encoder, panoptic segmentation decoder, and depth estimation decoder.

Image encoder. We extract image features from the input image frame by the image encoder. To deal with various scales of the instances, the encoder outputs multi-scale features $F = [F_0, F_1, F_2, F_{-1}]$ using a backbone and a pixel decoder following [3, 8]. We use ResNet-50 [7] backbone and multi-scale deformable attention Transformer [24] as the pixel decoder. We expect the shared encoder to learn and exploit mutual information of both segmentation and depth branches.

Panoptic segmentation decoder. We guide the query to learn the attributes, *e.g.*, appearance or location, of the objects in the panoptic segmentation decoder. The initial queries $Q_{seg} \in \mathbb{R}^{N\times C}$ and the image features F from the encoder are taken as input of the decoder. Note that the decoder is composed of three layers and each scale of feature except for F_{-1} is taken to each decoder layer. The queries are updated by computing cross-/self-attention with the given features at each layer. We do not intentionally split stuff queries from N queries so that each query will naturally indicate stuff of things instance by our training scheme.

Then the queries are passed to the two independent prediction heads: the classification head and the mask head. The queries are decoded into the class prediction $p(c) \in \mathbb{R}^{N\times(K+1)}$ and the mask embeddings $\mathcal{E}^m$ by each head. For each instance i, the instance mask is computed by dot-product between the mask embeddings and the last image feature as $m_i(h,w) = \sigma([\mathcal{E}_i^m]^\top F_{-1}(h,w))$, where $\sigma(\cdot)$ is the sigmoid function. We match the queries with the ground-truth instances using bipartite matching with the matching cost following [8]. At inference time, we merge all the instance masks into a single panoptic segmentation map as $m = \sum_{i=1}^{K} m_i$.

Depth estimation decoder. We adopt the depth decoder similar to the panoptic segmentation decoder. In the depth estimation decoder, the depth queries $Q_{depth} \in \mathbb{R}^{(N+1)\times C}$ are updated with the image features. Note that we utilize an additional query, *i.e.*, global depth query, to predict whole depth values of all pixels. Since the depth queries exploit same bipartite matching results from the segmentation decoder, only K depth queries are mapped to the ground-truth depth instances. Therefore, non-object regions, *e.g.*, gaps between objects with negligible confidence scores or unseen class objects, could not be handled with K matched depth queries and have undefined depth values. We thus add another query to N depth queries to fill in the values and make dense depth prediction.

After updated by the transformer decoder, the depth queries are decoded into the depth embeddings $\mathcal{E}^d$ by the depth prediction head $\mathcal{D}$. Each instance depth map is computed by dot-product between the depth embeddings and the image feature as $d_i(h,w) = \sigma([\mathcal{E}_i^d]^\top F_{-1}(h,w))$. The final prediction d is obtained by merging the instance depth maps d_i and the global depth map d_g with corresponding ground-truth instance mask $m_{gt,i} \in \{0,1\}^{H\times W}$ as

$$D = (1 - m_{gt}) \odot d_g + \sum_{i=1}^{K} m_{gt,i} \odot d_i, \qquad (4)$$

where $m_{gt} = \sum_{i=1}^{K} m_{gt,i}$, $m_{gt} \in \{0,1\}^{H\times W}$ refers to a single binary mask after merging the instance masks and hence $(1 - m_{gt}) \odot d_g$ term denotes the non-object regions to fill in the depth with predicted global depth map.

3.4. Object tracking at inference

After training the model, our model performs object tracking in a frame-by-frame manner. For example, if the model feed-forwards frame t and frame $t+1$ independently, the queries from each frame are utilized as a tool for linking frames. We associate the instances by bipartite matching between segmentation queries Q_{seg} for frame t and queries for frame $t+1$. We compute the cosine similarity of queries for the matching cost. This tracking by query matching is effective for well-performed shown in Fig. 3.

4. Experiments

4.1. Experiment Setup

Datasets. The Cityscapes-DVPS dataset [14] we employ is an extension of the Cityscapes-VPS dataset [10]. It enhances the dataset by introducing additional depth annotations, derived from the stereo disparity maps provided in the original Cityscapes dataset. Within the Cityscapes-DVPS dataset, there are 19 semantic classes, consisting of 8 'thing' classes and 11 'stuff' classes. The dataset comprises a total of 3,000 annotated frames. Specifically, the training, validation, and test sets contain 2,400, 300, and 300 frames, respectively. Note that we exclude 2 frames with problematic depth maps from the training data.

Evaluation metrics. In accordance with the established evaluation protocol, the evaluation results are analyzed by using the Depth-aware Video Panoptic Quality (DVPQ) metric following ViP-Deeplab [14]. This metric gauges the quality of video panoptic segmentation by focusing on pixels where the absolute relative depth error remains below a predetermined threshold value. To be more specific, we use the symbols P and Q to represent the prediction and ground-truth, respectively. We also let k be the window size, and λ be the depth threshold. We denote P_i^c, P_i^{id}, and P_i^d as the predictions of example i for the semantic class, instance ID, and depth. These notations are also applied to the corresponding ground-truth; Q_i^c, Q_i^{id}, and Q_i^d. Then, we can define DVPQ_λ^k as: $\text{PQ}\left(\left[\|_{i=t}^{t+k-1}(\hat{P}_i^c, P_i^{id}), \|_{i=t}^{t+k-1}\left(Q_i^c, Q_i^{id}\right)\right]_{t=1}^{T-k+1}\right)$. $\|_{i=t}^{t+k-1}(\cdot,\cdot)$ represents the concatenation of the pair of elements horizontally, specifically from t to $t+k-1$. As mentioned above, we exclude pixels that exceed the threshold value of absolute relative depth error. We measure four values of window size $k \in \{1,2,3,4\}$ and three values of threshold $\lambda \in \{0.1, 0.25, 0.5\}$ for Cityscapes-DVPS dataset following [14].

4.2. Results

Quantitative results. We compare our model with the competing method [23] on Cityscapes-DVPS [14] dataset in Table 1. Using the same ResNet-50 backbone [7], our model outperforms PolyphonicFormer on DVPQ and DVPQ-Stuff by 2.8%p and 5.0%p, respectively. Our method still achieves comparable results in DVPQ-thing. We postulate that per-frame learning is effective for depth-aware video panoptic segmentation. As k denotes the number of frames, the performance gaps across k frames imply the temporal consistency of successive frames. Without any temporal cues in training, our model achieves smaller performance drop between $\text{DVPQ}^{k=1}$ and $\text{DVPQ}^{k=4}$ than the compared method.

Qualitative results. We visualize the predictions of panoptic segmentation and depth estimation on two samples of consecutive frames in Cityscapes-DVPS datasets. As shown in Fig. 3, MinDVPS simultaneously predicts temporally consistent panoptic segmentation and depth estimation results without any temporal losses at training time and tracking modules at inference time. Our method also consistently tracks occluded instances in a video (*e.g.*, a car behind the other car and the person behind the pole), since the tracking is performed only via bipartite matching between $N \times C$ instance query embeddings, which do not have spatial extents [8].

Method	$k=1$			$k=2$			$k=3$			$k=4$			DVPQ$^k_\lambda$ Average		
MinDVPS (ours) $\lambda=0.50$	64.6	53.1	73.1	58.7	42.6	70.3	55.5	37.2	68.9	53.6	33.4	68.3	58.1	41.6	70.1
MinDVPS (ours) $\lambda=0.25$	61.5	50.5	69.6	55.3	39.4	66.9	53.0	36.0	65.4	51.0	32.4	64.5	55.2	39.6	66.6
MinDVPS (ours) $\lambda=0.10$	44.1	33.7	51.7	39.5	25.7	49.6	37.8	22.3	49.0	36.2	19.8	48.2	39.4	25.4	49.6
MinDVPS (ours) Avg.	56.8	45.8	64.8	51.2	35.9	62.3	48.8	31.9	61.0	46.9	28.6	60.3	**50.9**	35.5	**62.1**
PolyphonicFormer [23] Avg.	54.4	47.0	59.8	48.1	35.9	57.0	45.5	30.9	56.2	44.1	28.6	55.4	48.1	**35.6**	57.1

Table 1. **Quantitative results on Cityscapes-DVPS** Each cell shows | DVPQ$^k_\lambda$ DVPQ$^k_\lambda$-Thing DVPQ$^k_\lambda$-Stuff | where λ is the threshold of the relative depth error and k is the number of frames. Smaller λ and larger k requires the higher accuracy. Avg. denotes the averaged performances over the depth errors.

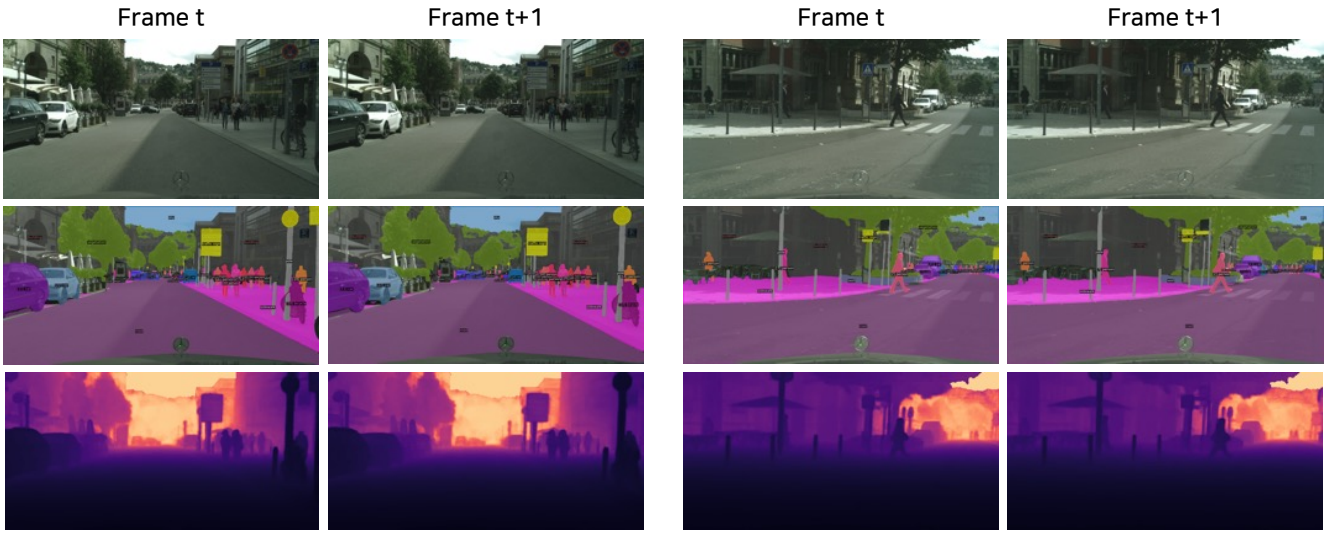

Figure 3. **Qualitative results on Cityscapes-DVPS**. We visualize the panoptic segmentation and depth prediction results of our model for two examples of consecutive image frames. The instances with the same color over two consecutive frames in the panoptic segmentation map denote that they have same instance ID. The brighter color in the depth map correspond to the higher value of depth. Note that we post-process the depth regions predicted as "sky" stuff by panoptic segmentation to have max depth value.

5. Conclusion

We present MinDVPS, a minimal model for depth-aware video panoptic segmentation. Without any video annotations, our model is trained with frame-level annotations just as image-based network. Instead of adding extra tracking modules, we process online tracking by well-learned queries and matching them frame-by-frame manner. MinDVPS outperforms previous work and shows temporally consistent performances across frames. This effective and efficient model for DVPS is helpful for autonomous driving system and also sports video assistant referee system where needs robust segmenting and tracking performances.

Acknowledgment. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub). This research project was supported by the Sports Promotion Fund of Seoul Olympic Sports Promotion Foundation from Ministry of Culture, Sports and Tourism.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[3] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask trans-

former for universal image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[4] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. 2014. 3

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 3

[6] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 4

[8] De-An Huang, Zhiding Yu, and Anima Anandkumar. Minvis: A minimal video instance segmentation framework without video-based training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3, 4

[9] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[10] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[11] Dahun Kim, Jun Xie, Huiyu Wang, Siyuan Qiao, Qihang Yu, Hong-Seok Kim, Hartwig Adam, In So Kweon, and Liang-Chieh Chen. Tubeformer-deeplab: Video mask transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[12] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *International Conference on 3D Vision (3DV)*, 2016. 3

[13] Andra Petrovai and Sergiu Nedevschi. Monodvps: A self-supervised monocular depth estimation approach to depth-aware video panoptic segmentation. In *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2023. 2

[14] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 4

[15] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1

[16] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[17] Sanghyun Woo, Dahun Kim, Joon-Young Lee, and In So Kweon. Learning to associate every segment for video panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[18] Junfeng Wu, Yi Jiang, Song Bai, Wenqing Zhang, and Xiang Bai. Seqformer: Sequential transformer for video instance segmentation. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[19] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *arXiv preprint arXiv:2211.05783*, 2022. 1

[20] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[21] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover learning for fast online video instance segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[22] Kim Youwang, Kim Ji-Yeon, Kyungdon Joo, and Tae-Hyun Oh. Unified 3d mesh recovery of humans and animals by learning animal exercise. *British Machine Vision Conference (BMVC)*, 2021. 1

[23] Haobo Yuan, Xiangtai Li, Yibo Yang, Guangliang Cheng, Jing Zhang, Yunhai Tong, Lefei Zhang, and Dacheng Tao. Polyphonicformer: Unified query learning for depth-aware video panoptic segmentation. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 4, 5

[24] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021. 3