Dense Pixel-level Interpretation of Dynamic Scenes with Video Panoptic Segmentation

Dahun Kim, *Member, IEEE,* Sanghyun Woo, *Student Member, IEEE,*
Joon-Young Lee, *Member, IEEE,* and In So Kweon, *Member, IEEE*

***Abstract*—A holistic understanding of dynamic scenes is of fundamental importance in real-world computer vision problems such as autonomous driving, augmented reality and spatio-temporal reasoning. In this paper, we propose a new computer vision benchmark: Video Panoptic Segmentation (VPS). To study this important problem, we present two datasets, Cityscapes-VPS and VIPER together with a new evaluation metric, video panoptic quality (VPQ). We also propose VPSNet++, an advanced video panoptic segmentation network, which simultaneously performs classification, detection, segmentation, and tracking of all identities in videos. Specifically, VPSNet++ builds upon a top-down panoptic segmentation network by adding pixel-level feature fusion head and object-level association head. The former temporally augments the pixel features while the latter performs object tracking. Furthermore, we propose panoptic boundary learning as an auxiliary task, and instance discrimination learning which learns spatio-temporally clustered pixel embedding for individual thing or stuff regions, *i.e.*, exactly the objective of the video panoptic segmentation problem. Our VPSNet++ significantly outperforms the default VPSNet, *i.e.*, FuseTrack baseline, and achieves state-of-the-art results on both Cityscapes-VPS and VIPER datasets. The datasets, metric, and models are publicly available at https://github.com/mcahny/vps.**

***Index Terms*—video panoptic segmentation, panoptic segmentation, video instance segmentation, video semantic segmentation, scene parsing.**

I. Introduction

DENSE and pixel-level interpretation of dynamic scenes is critical for real-world vision problems such as autonomous driving, augmented reality, and spatio-temporal reasoning. It requires tackling multiple tasks simultaneously to classify, detect, segment, and track all the scene elements. Solving these individual tasks provides a complementary interpretation of the scene. For example, semantic segmentation helps understand the context of surroundings, and instance segmentation and tracking present every dynamic object's temporal evolution in a scene. As an effort to unify these recognition tasks and leverage their mutual benefits, Kirillov *et al.* [1] proposed the *panoptic segmentation*, and a large number of approaches [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] have been proposed since then to this new benchmark, confirming its importance to the field.

In this paper, we extend panoptic segmentation into the video domain. This task requires assigning semantic classes and tracking id tags to all pixels and objects in a video. Fig. 1 illustrates a sample sequence of video panoptic segmentation ground truths. In panoptic segmentation, all the scene elements can be grouped into either 'things' or 'stuff' classes where the former denotes countable object instances and the latter denotes amorphous and non-countable regions. The task can be considered a simultaneous video segmentation of both things and stuff classes. Naturally, we name the new task *video panoptic segmentation* (VPS) [14]. As a pioneer work, solving the VPS task introduces three major challenges: the dataset, model architecture, and evaluation. We present our contributions that tackle these challenges throughout the paper.

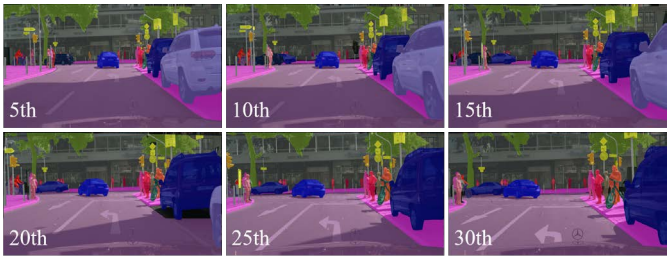


Fig. 1: Example video sequences of created Cityscapes-VPS annotations for video panoptic segmentation.

Dataset. Thanks to the existence of panoptic segmentation benchmarks such as COCO [15], Cityscapes [16], and Mapillary [17], the panoptic *image* segmentation has successfully driven active participation of the community. However, the direction towards the video domain has not yet been explored, probably due to the lack of appropriate datasets and evaluation metrics. While video object/instance segmentation datasets are available these days, no dataset permits direct training of video *panoptic* segmentation (VPS). This is not surprising when considering its extremely high cost of collecting such data. To improve the situation, we make an important first step in panoptic *video* segmentation by presenting two types (one real and one synthetic) of datasets. The first and primary dataset is *Cityscapes-VPS* that extends the public Cityscapes dataset to a video level. We sample every five video frames of a video and add panoptic segmentation annotations that are temporally consistent to the public image-level annotations. We consider this as our primary dataset as it is real data and enables a smooth image-to-video transition from the Cityscapes benchmark [16]. We provide profound statistics on the label distribution and track lengths of the dataset.

D. Kim, S. Woo and I.S. Kweon are KAIST, Daejeon, Korea.
E-mail: {mcahny, shwoo93, iskweon77}@kaist.ac.kr
J. Lee is with Adobe Research, San Jose, CA, USA.
E-mail: jolee@adobe.com
Manuscript received XXXX XX, XXXX; revised XXXX XX, XXXX.

Second, we adapt the synthetic VIPER [18] dataset into the video panoptic segmentation format and create corresponding metadata.

Model architecture. We propose VPSNet++ as an advanced video panoptic segmentation network. We build upon a top-down video panoptic segmentation network [14], which is built on top of the two-stage detector Mask R-CNN [19] and the panoptic segmentation network UPSNet [5]. To deal with video context, our VPSNet++ takes an additional reference frame sampled from a temporal neighbor of a target frame. Two main components are introduced to learn temporal correspondences among all the pixels and objects in the two frames, *i.e.*, pixel-level feature fusion head and object tracking head. First, the fusion operates bi-directionally between the reference and target frames and mutually augments their pixel features. Second, the tracking head is added to learn object association between the two frames based on their regional (RoI) feature similarity. Both fusion and tracking heads improve over the default VPSNet-FuseTrack baseline [14]. We further propose novel learning objectives: panoptic edge learning and spatio-temporal pixel embedding learning that encourages the video pixels from individual identity to be similar and those from different identities to be distinct, which is exactly the property required for video panoptic segmentation. As a result. VPSNet++ achieves the state-of-the-art results by outperforming the VPSNet baseline by +2.1% VPQ on VIPER and +1.4% VPQ on Cityscapes-VPS datasets.

Evaluation. We adapt the standard image panoptic quality (PQ) measure to fit the video panoptic quality (VPQ) format. Specifically, the metric is computed over a span of several frames, where a sequence of same-identity segments is considered a single spatio-temporal *tube* prediction. The predicted tubes are then matched to the ground truth tubes to compute their IoUs. The longer the time-span, the more challenging it is to obtain IoU over a threshold and to be counted as a true-positive when computing the VPQ score. We evaluate our proposed method with several other naive baselines using the VPQ metric.

Experimental results demonstrate that VPSNet without its tracking head can achieve state-of-the-art image-PQ on the Cityscapes benchmark. More importantly, our full VPSNet++ achieves state-of-the-art VPQ results on the Cityscapes-VPS and VIPER datasets.

We summarize the contribution of this paper as follows.

- This is a pioneer work that formally defines and studies the video panoptic segmentation (VPS) problem.
- We present spatially and temporally dense annotated datasets Cityscapes-VPS and a synthetic VIPER dataset, providing challenging segmentation and tracking of dynamic scenes.
- We propose a video panoptic quality (VPQ) metric that evaluates 3D overlap between the predicted and ground truths segments in both spatial and temporal dimensions.
- We propose VPSNet++ which improves the default VPSNet [14] with improved fusion and tracking heads and novel proposed learning objectives - panoptic edge learning and spatio-temporal identity discrimination learning.
- Our VPSNet++ outperforms strong image and video panoptic segmentation baselines on panoptic quality (PQ) and video panoptic quality (VPQ) metrics.

II. Related Work

A. Panoptic Segmentation

The joint task of thing and stuff segmentation is reinvented by Kirillov *et al.* [1] by combining the semantic segmentation and instance segmentation tasks and is named panoptic segmentation. Since then, much research [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] has been actively gathered to propose new approaches to this unified task. Recent approaches present end-to-end methods which can be grouped into two types: top-down and bottom-up methods. Top-down methods [4], [9], [6], [3], [5], [10] consist in two-stage approach which generates object proposals followed by the region-based prediction. Integrating Mask R-CNN [19] and panoptic FPN features [3], these methods show strength in capturing objects. Li *et al.* [9] suggest to enforce consistency between things and stuff pixels when merging them into a single segmentation result. Liu *et al.* [6] design a spatial ranking module to address the occlusion between the predicted instances. Xiong *et al.* [5] introduce a non-parametric panoptic head to predict instance id and resolve the conflicts between things and stuff segmentation. Bottom-up panoptic segmentation methods group pixels to form instances on top of semantic segmentation prediction [13], [20], [21]. For example, SSAP [22] learns pixel-pixel affinity pyramid and Panoptic-DeepLab [23] uses instance center regression on top of semantic segmentation prediction from DeepLab [24]. As opposed to top-down models, bottom-up panoptic segmentation models are advantageous at achieving high 'stuff' segmentation, but low with 'things'.

B. Video Semantic Segmentation

As a direct extension of semantic segmentation to videos, all pixels in a video are predicted as different semantic classes. However, the research in this field has not gained much attention and not currently popular compared to its counterpart in the image domain. One possible reason is the lack of available training data with temporally dense annotation, as research progress depends greatly on the existence of datasets. Despite the absence of a dataset for Video Semantic Segmentation (VSS), several approaches have been proposed in the literature [25], [26], [27], [28], [29]. Temporal information is utilized via optical flow to improve the accuracy or efficiency of the scene labeling performance. Different from our setting, VSS does not require either discriminating object instances or explicit tracking of the objects across frames. Our new *Cityscapes-VPS* is a super-set of a VSS dataset and thus is able to benefit this independent field as well.

C. Video Instance Segmentation

The Video Instance Segmentation (VIS) *et al.* [30] task requires tackling several multi-tasks: video object segmentation [31], [32], [33], [34], [35], [36], [37], [38] and video

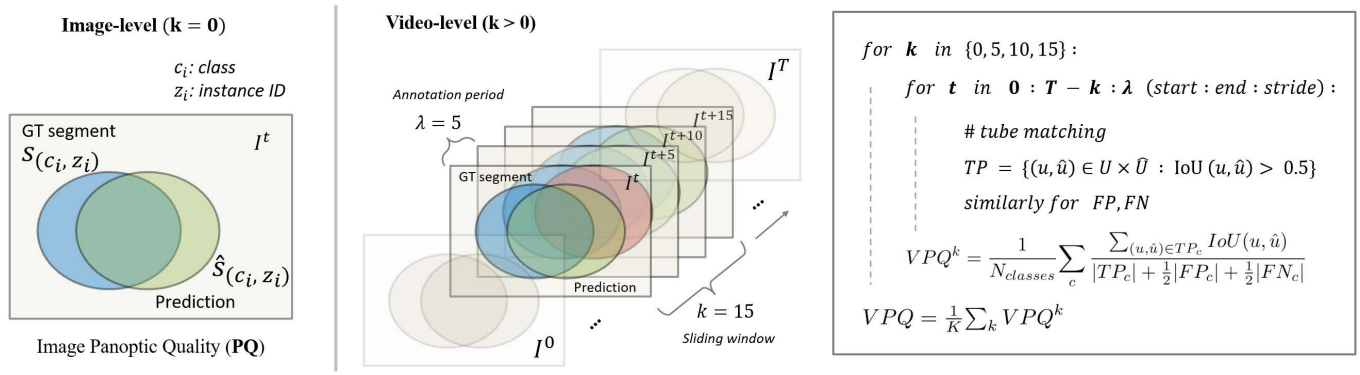


Fig. 2: **Tube matching and video panoptic quality (VPQ) metric.** An IoU is obtained by matching predicted and ground truth *tubes*. A frame-level false positive segment penalizes the whole predicted tube to get a low IoU. Each VPQ^k is computed by sliding the window through a video, and averaged by the number of frames. k indicate the temporal window size. VPQ^k is then averaged over different k values, to get a final VPQ score.

object detection [39], [40], [27], and aims at simultaneous detection, segmentation, and tracking of instances in videos. This task is also known as multi-object tracking and segmentation (MOTS) [41]. However, the focus of MOTS is on long videos with fewer number of target classes, *e.g.*, only *cars* and *pedestrians*, while VIS handles wider range of thing classes. As we aim at handling more general object (8 - 10 thing classes), we mainly consider VIS closely related to our work.

Yang et al [30] propose MaskTrack R-CNN that extends the Mask R-CNN [19] with a tracking branch and external memory that saves the features of instances across multiple frames. MaskProp [42] learns to reuse the predicted masks from neighbor frames to crop the extracted features, and temporally propagate the features to improve the segmentation and tracking. STEm-Seg [43] proposes to model video clips as spatial-temporal volumes and then separates object instances by learning to cluster the pixel embeddings.

In contrast to our Video Panoptic Segmentation task, VIS only deals with foreground *thing* objects but not background *stuff* regions. Moreover, the task permits overlaps between predicted object masks and even multiple predictions for a single instance, while our task requires algorithms to assign a single label to all things and stuff pixels. Last but not least, the main benchmarks dataset for VIS task is Youtube-VIS [30], but it contains a small number of objects (~ 5) per frame. In contrast, we deal with a much larger number of objects (> 20 on average), which makes our task even more challenging.

III. Problem Definition

A. Task Format

For a video sequence with T frames, we set a temporal window that spans additional k consecutive frames. Given a k-span snippet $I^{t:t+k} = \{I^t, I^{t+1}, ..., I^{t+k}\}$, we define a *tube* prediction as a track of its frame-level segments as $\hat{u}_{(c_i,z_i)} = \{\hat{s}^t, ..., \hat{s}^{t+k}\}_{(c_i,z_i)}$, for semantic class c and instance id z of the tube. Note that instance id z_i for a *thing* class can be larger than 0, *e.g.*, *car*-0, *car*-1, ... , whereas it is always 0 for a *stuff* class, *e.g.*, *sky*. All pixels in the video are grouped by such tuple prediction, and they will result in a set of *stuff* and *things* video tubes that are mutually exclusive to each other. The ground truth tube is defined similarly, with a slight adjustment concerning the annotation frequency as described below. The goal of video panoptic segmentation is to accurately localize all the semantic and instance boundaries throughout a video and assign correct labels to those segmented video tubes.

B. Evaluation Metric

By the construction of the VPS problem, no overlaps are possible among video tubes. Thus, AP metric used in object detection or segmentation cannot be used to evaluate the VPS task. Instead, we borrow the panoptic quality (PQ) metric in image panoptic segmentation with modifications adapted to our new task.

Given a snippet $I^{t:t+k}$, we denote a ***set*** of the ground truth and predicted tubes as $\mathcal{U}^{t:t+k}$ and $\hat{\mathcal{U}}^{t:t+k}$. A set of True Positive matches is defined as TP = $\{(u, \hat{u}) \in \mathcal{U} \times \hat{\mathcal{U}} : \text{IoU}(u, \hat{u}) > 0.5\}$. False Positives (FP) and False Negatives (FN) are defined accordingly. When the annotation is given every λ frames, the matching only considers the annotated frame indices $t : t + k : \lambda$ $(start : end : stride)$ in a snippet, *e.g.*, when k = 10 and λ = 5, frame t, t+5 and t+10 are considered. We slide the k-span window with a stride λ throughout a video, starting from frame 0 to the end, *i.e.*, t goes by $0 : T - k : \lambda$ (We assume frame 0 is annotated). Each stride constructs a new snippet, where we compute the IoUs, TP, FP and FN as above.

At a dataset level, the snippet-level IoU, |TP|, |FP| and |FN| values are collected *across all predicted videos*. Then, the *dataset-level* VPQ metric is computed per each class c, and averaged across all classes as,

$$VPQ^k = \frac{1}{N_{classes}} \sum_c \frac{\sum_{(u,\hat{u}) \in TP_c} IoU(u, \hat{u})}{|TP_c| + \frac{1}{2}|FP_c| + \frac{1}{2}|FN_c|}, \quad (1)$$

where $\frac{1}{2}|FP| + \frac{1}{2}|FN|$ in the denominator is to penalize unmatched tubes, as suggested in the image PQ metric.

By definition, k = 0 will make the metric equivalent to the image PQ metric, and k = T-1 will construct a set of whole video-long tubes. Any cross-frame inconsistency of semantic or instance label prediction will result in a low tube IoU, and may drop the match out of the TP set, as illustrated in Fig. 2. Therefore, the larger window size we have, the more challenging it is to get a high VPQ score. In practice, we include different window sizes $k \in \{0, 5, 10, 15\}$ to provide a more comprehensive evaluation. The final VPQ is computed by averaging over K = 4 as, $VPQ = \frac{1}{K}\sum_k VPQ^k$.

Having different k values enables a smooth transition from the existing image PQ evaluation to videos, encouraging the image-to-video transition of further technical developments for this pioneering field to leap forward.

C. Hyper-parameter: Temporal Window

We set k as a user-defined parameter. Having such a fixed temporal window size regularizes the difficulty of IoU matching across video samples of different lengths. On the other hand, the difficulty of matching whole T-long tubes, largely varies with the video length, *e.g.*, when T = 10 and T = 1000.

We empirically observed that, in our Cityscapes-VPS dataset ($\lambda = 5$), many object associations are disconnected by significant scene changes when $k > 15$. Given a new annotation frequency (1/λ), the k shall be reset, which will accordingly set a level of difficulty for the dataset.

IV. Dataset Collection

A. Existing Image-level Benchmarks

There are several public datasets which have dense panoptic segmentation annotations: Cityscapes [16], ADE20k [44], Mapillary [17], and COCO [15]. However, none of these datasets matches the requirement for our video panoptic segmentation task. Thus, we need to prepare a suitable dataset for the development and evaluation of video panoptic segmentation methods. We pursue several directions when collecting VPS datasets. First, both the quality and quantity of the annotation should be high, of which the former is a common problem in some of the existing polygon-based segmentation datasets and the latter is limited by the extreme cost of panoptic annotations. More importantly, it should be easily adaptable to and extensible from the existing image-based panoptic datasets, so that it can promote the research community to seamlessly transfer the knowledge between the image and video domains. With the above directions in mind, we present two VPS datasets by 1) creating new Cityscapes-VPS dataset that adds video panoptic segmentation annotations based on the Cityscapes dataset and 2) reformatting the VIPER dataset.

B. Cityscapes-VPS

Instead of building our dataset from scratch in isolation, we build our benchmark on top of the public Cityscapes dataset [16], which is the most popular dataset for panoptic segmentation, together with COCO. It consists of image-level annotated frames of ego-centric driving scenarios, where each labeled frame is the 20th frame in a 30 frame video snippet. There are 2965, 500, and 1525 such sampled images paired with dense panoptic annotations for 8 *thing* and 11 *stuff* classes for training, validation, and testing, respectively. Specifically, we select the validation set to build our own video-level extended dataset. We select every five frames from each of the 500 videos, *i.e.*, 5, 10, 15, 20, 25, and 30-th frames, where the 20-th frame already has the original Cityscapes panoptic annotations. For the other 5 frames, we ask expert turkers to carefully label each pixels with all 19 classes and instance ids to be consistent with the 20-th frame as reference. It is also asked to have similar level of pixel quality, as shown in Fig. 1-(bottom row). Our resulting dataset provides additional 2500 frames of dense panoptic labels at 1024×2048 resolution that temporally extend the 500 frames of the Cityscapes labels. The new benchmark is referred to as *Cityscapes-VPS*.

Our new dataset *Cityscapes-VPS* is not only the first benchmark for video panoptic segmentation but also a useful benchmark for other vision tasks such as video instance segmentation and video semantic segmentation; the latter has also been suffering lack of well-established video benchmark.

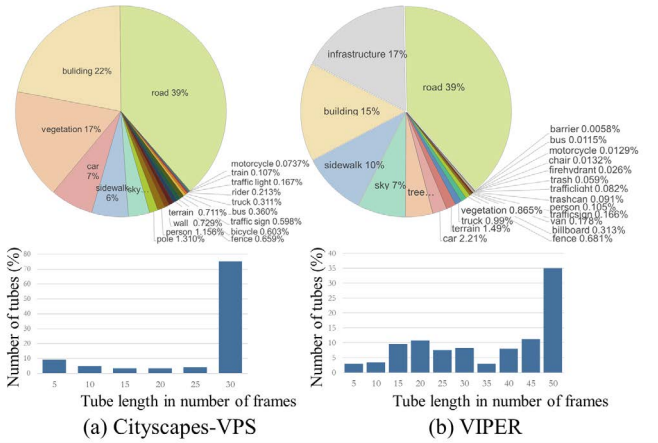


Fig. 3: Label distribution and track length in Cityscapes-VPS and reformatted VIPER datasets.

	YT-VIS	City	re-VIPER	City-VPS
Videos	2540	3475	124	500
Frames	108k	3475	184k	3000
Things	40	8	10	8
Stuff	x	11	13	11
Instances	4297	60 K	31 K	10 K
Masks	115 K	60 K	2.8 M	56 K
Temporal	✓	x	✓	✓
Dense (Panoptic)	x	✓	✓	✓

TABLE I: High-level statistics of our reformatted VIPER and new Cityscapes-VPS (additional to the original Cityscapes) with previous video instance / semantic segmentation datasets. YT-VIS and City stands for YouTube-VIS and Cityscapes respectively.

C. Revisiting VIPER dataset

To maximize both the quality and quantity of the available annotations for the VPS task, we take advantage of the synthetic VIPER dataset [18] extracted from the GTA-V game engine. It includes pixel-wise annotations of semantic and instance segmentations for 10 *thing* and 13 *stuff* classes on 254K frames of ego-centric driving scenes at 1080×1920 resolution. As shown in Fig. 1-(top row), we tailor their annotations into our VPS format and create metadata in a popular COCO style, so that it can be seamlessly plugged into recent recognition models such as Mask-RCNN [19].

D. Dataset Statistics

We show some high-level statistics of the Cityscapes-VPS, reformatted VIPER, and related datasets in Table. I. Also, we illustrate the class-wise histograms, *i.e.*, amount of pixels in the dataset in Fig. 3. As shown in the figure, both for Cityscapes-VPS and VIPER, 'cars' and 'persons' contain more pixels compared to the rest of the classes. We also show the histograms for tube (tracklet) lengths in each dataset. As shown in the figure, most tracklet tubes in both Cityscapes-VPS and VIPER datasets are of the whole snippet length, *i.e.*, the instance appears from first to last frame of a clip. This presents a challenge for long term consistency in segmentation and tracking.

V. Video Panoptic Segmentation Network

We propose VPSNet++, and advanced video panoptic segmentation network which can simultaneously perform panoptic segmentation and tracking of instances. In this section, we present the network architecture and its training losses and inference details.

A. Baselines

We build our solution on top of the top-down video panoptic segmentation network VPSNet [14], which is built on top of the two-stage detector Mask R-CNN [19] and the panoptic segmentation network UPSNet [5]. VPSNet-Track adds a Mask-Track head [30] to learn the correspondence between the instances from different frames based on their regional feature similarity. VPSNet-FuseTrack further adds a module for learning feature map flow between frames and attentional space-time feature fusion. Our VPSNet++ improves over the above baselines.

B. Modification in Image Panoptic Segmentation Network

Our image-level panoptic segmentation network is based on UPSNet [5]. We add an extra non-parametric layer at the feature pyramid, which is inspired by Pang *et al.* [45]. They use *balanced semantic features* to enhance the pyramidal neck representations. Different from them, our main design purpose is to compute a representative single-resolution feature map from the multi-scale feature maps. The single-resolution representation is used in the module of VPSNet++ which will be detailed in Sec. V-C. This is implemented by a *gather* operation, where the feature pyramid network (FPN) [46] features $\{p^2, p^3, p^4, p^5\}$ are resized to the highest resolution *i.e.*, size of p^2 or 1/4 image size, and element-wise summed into a single-resolution feature f. This comprehensive feature is *redistributed* to the multi scales to augment the original FPN features.

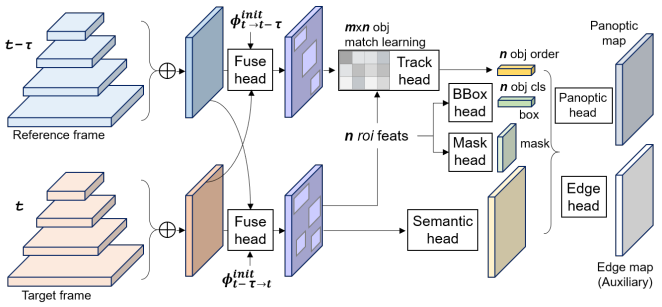


Fig. 4: **VPSNet++ architecture overview.** Based on an image-level panoptic segmentation network, our VPSNet++ learns the inter-frame correspondences in both pixel level and object level. First, Bi-directional Fuse head is proposed to propagate and augment pixel features between the reference and target frames via learnable feature flows. The temporally aggregated features are used in all downstream task branches. Second, the Contrastive Track head performs instance association by using the regional feature similarities between the frames. Finally, we add panoptic edge prediction learning as an auxiliary task during training. Our VPSNet++ simultaneously performs classification, detection, segmentation, and tracking (and optionally, edge prediction), *i.e.*, video panoptic segmentation.

C. VPSNet++ Model Architecture

An overview of our VPSNet++ is shown in Fig. 4. Our VPSNet++ takes a target frame I_t and a reference frame $I_{t-\tau}$. During training, the reference frame is sampled from a window $\tau \in \{-5 : +5\}$, while it is set to I_{t-1} (*i.e.*, previous frame) at testing. The target and reference images are independently processed through a CNN backbone and the balanced FPN (Sec. V-B). On top of this, we propose Bi-directional Fuse Head (Sec. V-C1) that learns to aggregate the pixel-level features between the two frames to enhance the feature maps. Then, we learn Contrastive Track Head with Hard Examples (Sec. V-C2) that predicts the correspondence between the instances from the two frames.

1) Bi-directional Fuse Head: Given the balanced pyramid feature maps from the reference and target frames $\{p^2, p^3, p^4, p^5\}_t$ and $\{p^2, p^3, p^4, p^5\}_{t-\tau}$, the pixel-level feature fusion is performed between their *gathered* feature maps f_t and $f_{t-\tau}$ (see Fig. 5). More specifically, we learn the bi-directional pixel-level correspondence between f_t and $f_{t-\tau}$ to augment the per-frame features. We adopt an *align-and-attend* pipeline which first aligns the feature maps using optical flow and merges them with spatio-temporal attention. For simplicity, we will describe the align-and-attend feature fusion from the reference to the target frame ($t \rightarrow t - \tau$). Note,

however, that the actual fusion operates in both directions ($t \rightarrow t-\tau$ and $t-\tau \rightarrow t$).

Align module is given an initial optical flow $\phi^{init}_{t \rightarrow t-\tau}$ computed by FlowNet2 [47]. As the pre-computed optical flow between I_t and $I_{t-\tau}$ might be sub-optimal for propagating representations across feature maps f_t and $f_{t-\tau}$, we introduce a shallow flownet that learns to refine the initial flow, guided by the final panoptic segmentation objectives. The shallow flownet takes the concatenation of the 2-channel initial flow $\phi^{init}_{t \rightarrow t-\tau}$, and the 256-channel target and reference feature maps (total 514 channels). It is composed of four 3×3 convolutional layers whose output channel size is 64, 64, 32, and 2, respectively. The final feature flow $\phi_{t \rightarrow t-\tau}$ is used to warp the reference feature onto the target feature, denoted by $f_{t-\tau \rightarrow t}$.

Attend module is given a pair of the spatially aligned features f_t and $f_{t-\tau \rightarrow t}$, and learns to re-weight and merge them into one. Inspired by [48], we first compute the pixel-wise frame similarity between f_t and $f_{t-\tau \rightarrow t}$ based on the per-pixel inner-product operation as

$$tAtt = \sigma(\epsilon(f_t) \cdot \epsilon(f_{t-\tau \rightarrow t})), \tag{2}$$

where σ is a sigmoid function and ϵ is an embedding layer with 3×3 kernel. The attention map $tAtt$ is of the same size with $f_{t-\tau \rightarrow t}$ and represents per-pixel confidence of the reference feature map. $tAtt$ and $f_{t-\tau \rightarrow t}$ are element-wise multiplied and concatenated with f_t along a temporal dimension resulting in a tensor shape $2 \times height \times width \times channel$. A following $2 \times 3 \times 3$ convolution layer reduces the time dimension into one which is the output feature of fuse head g_t. Finally, the target frame features $\{p^2, p^3, p^4, p^5\}_t$ are augmented by resizing-and-adding g_t to all multi-scale features, and these augmented features are fed into the downstream detection, segmentation, and tracking heads. Similarly, the reference frame features are augmented by the target frame features.

2) Contrastive Track Head with Hard Examples: Following the temporally augmented feature maps is a Track head that predicts the instance correspondences between the two frames.

The goal of Track head is to find object-object correspondences between two frames. Suppose there are M instances identified in the reference frame. Then new detected boxes from the target frame can only be associated with one of the M previous identities or assigned a new identity.

We formulate this as 1-*vs*-others discrimination problem. Specifically, the similarity is learned between each target RoI feature embedding r_i from I_t and the reference M+1 RoI feature embeddings $r_{j=0,...,M}$ from $I_{t-\tau}$ which represent the M already identified objects (j=1,...,M) and a new object which is denoted by a zero vector r_0.

We propose to learn such regional embeddings using the contrastive loss and hard example mining. Hard example mining is known to be crucial for embedding learning, as most examples are easy and do not contain much information to improve the model. For each target RoI embedding (anchor) r_i, we sample top-N_h negative samples from reference embeddings r_j by their distances to anchor. Since the number

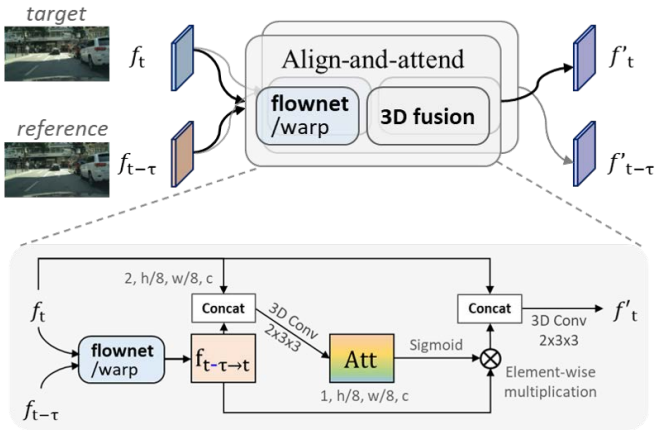


Fig. 5: **Bi-directional Fuse head.** The default Fuse head performs feature transformation in single direction from reference to target frame, which may lead to inconsistency between the two frame features. We propose bi-directional fusion between f_t and $f_{t-\tau}$ to obtain more balanced feature maps between different time steps.

of reference RoIs, M, is variable during training, we set $N_h = min(4, M)$. We use supervised contrastive learning [49] with noise contrastive estimation loss [50] as

$$L_{contra\_track} = -\sum_i log \frac{e^{A(f_i, f_{y_i})}}{e^{A(f_i, f_{y_i})} + C \sum_{j \in N_h} e^{A(f_i, f_j)}} \tag{3}$$

where C represents a weighting parameter, N_h the hard negative examples and A is defined as the cosine distance function:

$$A(r_i, r_j) := \frac{1}{\lambda} (\frac{r_i}{||r_i||})^T (\frac{r_j}{||r_j||}) \tag{4}$$

where λ is the temperature. Summation over hard examples allows the loss to focus on difficult cases and perform better. We use an external dictionary to store the reference RoI embeddings where we update or extend the memory when a new candidate box is assigned with an instance label.

The target and reference RoI feature embeddings are based on the enhanced features from the Bi-directional Fuse head, *i.e.*, g_t and $g_{t-\tau}$. Therefore, from a standpoint of the instance tracking, VPSNet++ synchronizes the tracking on both pixel-level and object-level. The pixel-level fusion aligns local feature of the instance to transfer it between the reference and target frames, and the object-level Track head focuses more on distinguishing the target instance from other reference objects by learning similarities among the temporally augmented RoI features.

During the inference stage, we add an additional cue coming from the panoptic head: the IoU of *things* logits. The IoU of instance logits can be viewed as a deformation factor or spatial correlation between frames and our experiments show that it improves the video panoptic quality for *things* classes.

3) Panoptic Edge Learning: Inspired by several prior works [51], [52] that train a model with both segmentation

masks and semantic/instance boundaries to improve the segmentation quality, we propose to learn a holistic panoptic boundary as an auxiliary task during training. The *panoptic boundary* prediction aims at differentiating all thing and stuff identities, and helps delineating the overlapping objects and complex scene elements. Moreover, the ground truth comes at no cost from the given panoptic segmentation annotations. We use the Laplacian operator to generate soft boundaries and threshold at 0 to convert them into a single ground-truth edge map S_t.

We construct the input tensor U_{edge} by combining semantic and instance segmentation logits. For stuff classes, we directly use the semantic segmentation logits of the corresponding channels. For any thing instance, we take its mask logits from the mask head which is of size 28×28, and interpolate back to the same scale $H \times W$ via bilinear interpolation and zero-padding outside its ground truth box. All stuff and thing logits are concatenated and are fed to the Edge head f_{edge} which consists of three 3×3 convolution layers with output channel size 16, 16 and 1, respectively. A sigmoid activation σ is appended at the last layer and the Edge head is trained with Dice loss [53]:

$$L_{pan\_edge} = Dice(S_t, \sigma(f_{edge}(U^t_{edge}))). \quad (5)$$

Note that the panoptic Edge head is only trained as an auxiliary task head, but not used during inference.

4) Panoptic Tube Id Discrimination Learning: Given the predicted panoptic feature map U^t, We use a per-pixel identity discrimination loss to help learning to discriminate different mask segments of all thing and stuff classes. To this end, we compute representative embedding vectors for all segments in a frame. Specifically, we use the ground truth thing and stuff masks to spatially pool the predicted panoptic feature map for all individual segments. This results in N_{seg} segment-level embeddings $\{u^t_i\}$. We also construct the segment-level embeddings at $t-\tau$, and only take those embeddings $\{u^{t-\tau}_j\}$ when segment j is stuff class, or has the same tracking ID with one of $\{u^t_i\}$ by using the ground truth tracking labels. These associated (tracked) embeddings are element-wise averaged, and now the merged representations are the tube-level embeddings $\{u^{t|t-\tau}_i\}$ (see Fig. 6).

We enforce each pixel feature $U^t_{h,w}$ to perform segment ID discrimination task where each pixel should correctly identify which segment embedding out of N_{seg} it corresponds to. We use the per-pixel contrastive loss as:

$$L_{tube\_disc} = -\sum_{h,w} log \frac{\sum_i m_{i,h,w} \cdot e^{(u^{t|t-\tau}_i \cdot U^t_{h,w})}}{\sum_i e^{(u^{t|t-\tau}_i \cdot U^t_{h,w})}}, \quad (6)$$

where $m_{i,h,w}$ is non-zero only when pixel (h,w) corresponds to the ground truth tube id i. The per-pixel loss is applied to all thing and stuff pixels in a frame t. The tube-level contrastive learning encourages features from the same tube identity to be similar (both spatially and temporally) and features from different tubes to be distinct, which aligns well with the aim of video panoptic segmentation.

Our panoptic tube id discrimination loss is inspired by previous works [54], [55], [49], [56], [57]. They discriminate only thing instances either unsupervisedly or within a single image. Unlike theirs, we perform a complete thing-and-stuff discrimination learning is spatial-temporal, both within a frame and across frames, which is exactly the property required for video panoptic segmentation. Also note that the tube id discrimination learning is only used during training, but not used at inference.

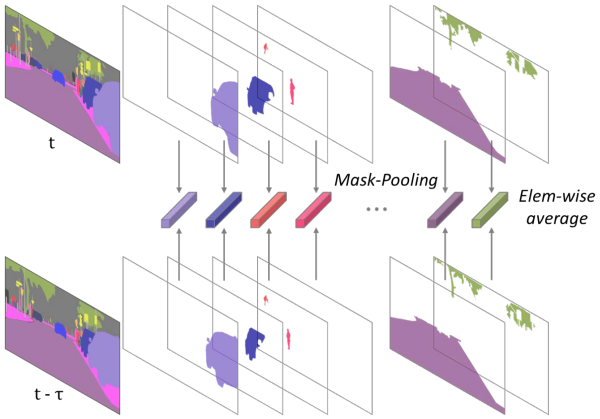


Fig. 6: **Tube-level embeddings.** We use the ground truth thing and stuff masks to spatially pool the predicted panoptic feature map for all individual segments. The mask-pooled features from two frames are element-wise averaged to construct the tube-level representation embeddings. These embeddings are used for the spatial-temporal pixel embedding learning, *i.e.*, panoptic tube id discrimination learning.

D. Implementation Details

We follow most of the settings and hyper-parameters of Mask R-CNN [19] and other panoptic segmentation models such as UPSNet [5]. Hereafter, we only explain those which are different. Throughout the experiments, we use ResNet-50 FPN [58], [46] as the feature extractor.

1) Training: We implement our models in PyTorch [59] with MMDetection [60] toolbox. We use the distributed training framework with 8 GPUs. Each mini-batch has 1 image per GPU. We use the ground truth box of a reference frame to train the track head. We crop random 800×1600 pixels out of 1024×2048 Cityscapes and 1080×1920 VIPER images after randomly scaling each frame by 0.8 to 1.25 $\times$. Due to the high resolution of images, we downsample the logits for semantic head and panoptic head to 200×400 pixels.

Besides the RPN losses, training of the image panoptic segmentation network (L_{ips}) contains loss functions for 3 task-related heads: bounding box head (classification and regression), mask head, and semantic head. The objectives of our VPSNet++ (L_{vps++}) contains additional losses for the proposed contrastive tracking (L_{contra_track}), panoptic edge learning (L_{pan_edge}) and tube-level discrimination learning (L_{tube_disc}). We set all loss weights to 1.0 to make their scales

Model variant	feat. align	feat. attend	obj. match	PQ	PQ^{Th}	PQ^{St}
Base				52.1	47.2	56.2
Align	✓			52.3	47.3	56.4
Attend		✓		50.7	45.8	54.8
Fuse	✓	✓		53.0	48.3	57.0
Track			✓	53.0	47.9	57.2
FuseTrack	✓	✓	✓	**55.4**	**52.2**	**58.0**

TABLE II: **Image panoptic segmentation results on VIPER.**

Method	Backbone	PQ	PQ^{Th}	PQ^{St}
AUNet [4]	ResNet-101	59.0	54.8	62.1
PanopticFPN [3]	ResNet-101	58.1	52.0	62.5
DeeperLab [13]	Xception-71	56.5	-	-
Seamless [10]	ResNet-50	59.8	54.6	63.6
AdaptIS [61]	ResNet-50	59.0	55.8	61.3
TASCNet [9]	ResNet-50	55.9	50.6	59.8
UPSNet [5]	ResNet-50	59.3	54.6	62.7
TASCNet+CO [9]	ResNet-50	59.2	56.0	61.5
UPSNet+CO [5]	ResNet-50	60.5	57.0	63.0
VPSNet-Base+CO	ResNet-50	60.6	57.0	63.2
VPSNet-Fuse+CO	ResNet-50	61.6	57.7	64.4
VPSNet-Fuse+VP	ResNet-50	**62.2**	**58.0**	**65.3**

TABLE III: **Image panoptic segmentation results on Cityscapes *val.* set.** '+CO' and '+VP' indicate the model is pretrained on COCO and VIPER, respectively.

to be roughly on the same order of magnitude:

$$L_{ips} = L_{class} + L_{box} + L_{mask} + L_{semantic} + L_{panoptic},$$
$$L_{vps++} = L_{ips} + L_{contra\_track} + L_{pan\_edge} + L_{tube\_disc}. \quad (7)$$

We set the learning rate and weight decay as 0.005 and 0.0001 for all datasets. For VIPER, we train for 12 epochs and apply lr decay at 8 and 11 epochs. For both Cityscapes and Cityscapes-VPS, we train for 144 epochs and apply lr decay at 96 and 128 epochs. For the pretrained models, we import COCO- or VIPER-pretrained *Base* model parameters and initialize the remaining layers, *e.g.*, Fuse (*align-and-attend*) head, Track head and Edge head, by Kaiming initialization.

2) Inference: Given a new testing video, our VPSNet++ processes each frame sequentially in an online fashion. At each frame, our VPSNet++ first generates a set of instance hypotheses. As a mask pruning process, we perform the class-agnostic non-maximum suppression with the box IoU threshold as 0.5 to filter out some redundant boxes. Then the remaining boxes are sorted by the predicted class probabilities and kept if the probability is larger than 0.6. For the first frame of a video sequence, we assign instance ids according to the order of the probability. For all other frames, the remaining boxes after pruning are matched to identified instances from previous frames based on the learned affinity A, and are assigned an instance id accordingly. After processing all frames, our method produces a sequence of panoptic segmentation outputs, each pixel of which contains a unique category label and instance id label throughout the sequence. For both image panoptic quality (IPQ) and video panoptic quality (VPQ) evaluation, we test all available models with single-scale testing.

VI. Experimental Results

In this section, we present the experimental results on the two proposed video-level datasets, *VIPER* and *Cityscapes-VPS*, as well as the conventional image-level Cityscapes benchmark. In particular, we mainly investigate the results in two aspects: image-level prediction and cross-frame association, which will be reflected in the IPQ and VPQ, respectively. We demonstrate the contributions of each component of VPSNet++. Here is the information on the datasets used in experiments.

- **VIPER**: Based on its high quantity and quality of the panoptic video annotation, we mainly experiment with this benchmark. We follow the public train / val split. For evaluation, we choose 12 validation videos from *day* scenario, and use the first 50 frames of each videos: total 600 images.
- **Cityscapes**: We use the public train / val split, and evaluate our image-level model on the validation set.
- **Cityscapes-VPS**: The created video panoptic annotations are given with the 500 validation videos of Cityscapes. We further split these videos into 400 training, 50 validation, and 50 test videos. Each video consists of 30 consecutive frames, with every 5 frames paired with the ground truth annotations. For each video, all 30 frames are predicted, and only the 6 frames with the ground truth are evaluated.

A. Image Panoptic Quality

Before delving into the video (spatial-temporal) quality of panoptic segmentation outputs, we first evaluate whether the VPS learning improves per-frame (spatial) panoptic quality. We use the existing panoptic quality (PQ), recognition quality (RQ), and segmentation quality (SQ) for the evaluation. The results are presented in Table. II and Table. III.

First, we study the importance of the proposed Fuse and Track modules to our image-level panoptic segmentation performance on the VIPER dataset as shown in Table. II. We find that both pixel-level and object-level modules have complementary contributions, each improving the baseline by +1% PQ. Without any of them, the PQ will drop by -3.4%. The best PQ was achieved when these two modules are used together.

We also experiment on the Cityscapes benchmark, to provide a comparison with the state-of-the-art panoptic segmentation methods. Note we can only 'Track' model cannot be trained in this setting, without tracking annotations in the Cityscapes dataset. Instead, we report 'Fuse' model results as it only requires a neighboring reference frame without any extra annotations. In Table. III, we find that our Fuse model outperforms the state-of-the-art baseline method [5] by +1.0% PQ, which implies that it effectively exploits spatial-temporal context to improve per-frame pixel features. The pretraining on the VIPER dataset shows its complementary effectiveness to either COCO or Cityscapes dataset by boosting the score by +1.6% PQ from our baseline, achieving 62.2% PQ. We also converted our results into *semantic* segmentation format, and achieved 79.0% mIoU.

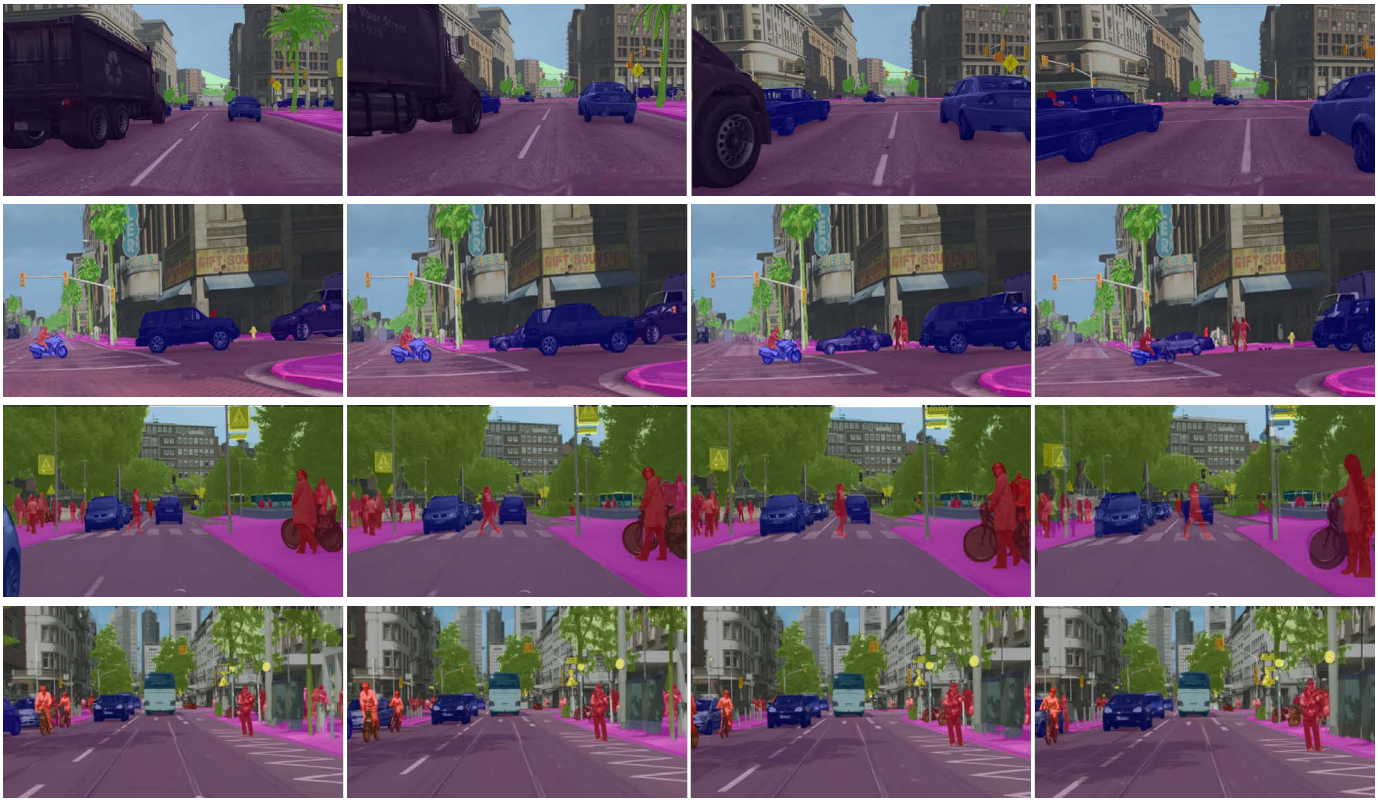


Fig. 7: **Video panoptic segmentation results of VPSNet++ on the VIPER (top two rows) and Cityscapes-VPS (bottom two rows) sequences.** Each row has four sampled frames from a video sequence. Objects with the same predicted identity have the same color.

Model variant	Tracking method	Improvement method	Temporal window size				VPQ
			k = 0	k = 5	k = 10	k = 15	
Track	All methods	-	48.1 / 38.0 / 57.1	49.3 / 45.6 / 53.7	45.9 / 37.9 / 52.7	43.2 / 33.6 / 51.6	46.6 / 39.0 / 53.8
FuseTrack	Cls-Sort	-	49.8 / 40.3 / 57.7	29.8 / 0.9 / 53.8	29.1 / 0.7 / 52.8	28.8 / 0.5 / 52.3	34.4 / 10.6 / 54.2
FuseTrack	IoU-Match	-	49.8 / 40.3 / 57.7	44.4 / 33.1 / 53.8	40.0 / 24.5 / 52.8	37.8 / 20.5 / 52.3	43.0 / 29.6 / 54.2
FuseTrack	Feat-Match	-	49.8 / 40.3 / 57.7	50.4 / 46.4 / 53.8	45.7 / 37.1 / 52.8	43.5 / 33.0 / 52.3	47.4 / 39.2 / 54.2
FuseTrack	All methods	-	49.8 / 40.3 / 57.7	51.6 / 49.0 / 53.8	47.2 / 40.4 / 52.8	45.1 / 36.5 / 52.3	48.4 / 41.6 / 54.2
VPSNet++	All methods	a. Bi-Fuse	50.5 / 40.8 / 58.6	52.9 / 50.8 / 54.8	48.5 / 42.3 / 53.4	46.4 / 38.6 / 52.8	49.5 / 43.1 / 54.9
VPSNet++	All methods	b. Contra-Track	49.8 / 40.4 / 57.7	52.7 / 51.2 / 53.8	48.0 / 42.1 / 52.8	45.7 / 37.7 / 52.4	49.1 / 42.9 / 54.2
VPSNet++	All methods	c. Pan-Edge	50.2 / 40.7 / 58.1	52.2 / 49.6 / 54.3	47.8 / 41.2 / 53.3	45.8 / 37.3 / 52.8	49.0 / 42.2 / 54.6
VPSNet++	All methods	d. Tube-Disc	50.7 / 41.1 / 58.7	53.2 / 51.0 / 55.0	48.7 / 42.8 / 53.6	46.4 / 38.6 / 52.8	49.8 / 43.6 / 55.0
VPSNet++	All methods	e. a + d	51.1 / 41.9 / 58.8	53.3 / 51.2 / 55.0	49.1 / 43.8 / 53.6	47.0 / 39.7 / 53.0	50.1 / 44.1 / 55.1
VPSNet++	All methods	f. a + b + c + d	**51.4** / 42.5 / 58.8	**53.7** / 52.0 / 55.1	**49.5** / 44.4 / 53.7	**47.4** / 40.5 / 53.1	**50.5** / 44.8 / 55.2

TABLE IV: **Video panoptic segmentation results on VIPER dataset.** Each cell contains VPQ / VPQ^{Th} / VPQ^{St} scores.

B. VPSNet++ Results

To demonstrate the effectiveness of our proposed VPSNet++, we conduct experiments on the VIPER and Cityscapes-VPS datasets. We evaluate the video panoptic quality (VPQ) scores and report them in Table. IV, Table. V and Table. VI. The visualization of video segmentation and flow refinement results of VPSNet++ is shown in Fig. 7 and Fig. 8.

Note that VPSNet++ is identical to the FuseTrack model except its improvement method(s) (denoted as a. - f. in the tables). Across all experiments, the mean VPQ of things classes (VPQ^{Th}) is generally lower than that of stuff classes (VPQ^{St}), as it is required extra consistency in instance ids over time. We also discuss ablation studies and visualization results.

1) Baseline Results: We present several baseline video panoptic segmentation methods. The baseline methods are the Track and FuseTrack variants of the default VPSNet. We first experiment on the VIPER dataset by enumerating different tracking methods: associating objects based on the distance between their classification logit values (Cls-Sort), flow-guided object matching by mask IoU (IoU-Match), and the RoI feature similarity based matching by the default Track head (Feat-Match). First, Cls-Sort relies on semantic consistency of the same object between frames. However, it fails to track objects possibly because there are a number of instances of the same class in a frame, *e.g.*, car, person, thus making the

Model variant	Improvement method	Temporal window size				VPQ
		k = 0	k = 5	k = 10	k = 15	
Track	-	63.1 / 56.4 / 68.0	56.1 / 44.1 / 64.9	53.1 / 39.0 / 63.4	51.3 / 35.4 / 62.9	55.9 / 43.7 / 64.8
FuseTrack	-	64.5 / 58.1 / 69.1	57.4 / 45.2 / 66.4	54.1 / 39.5 / 64.7	52.2 / 36.0 / 64.0	57.0 / 44.7 / 66.0
VPSNet++	a. Bi-Fuse	64.6 / 58.1 / 69.3	58.3 / 46.0 / 67.2	55.3 / 40.4 / 66.2	53.7 / 36.8 / 66.0	57.9 / 45.3 / 67.2
VPSNet++	b. Contra-Track	65.0 / 58.3 / 69.9	57.6 / 45.6 / 66.4	54.8 / 40.3 / 65.3	52.9 / 36.5 / 64.8	57.6 / 45.2 / 66.6
VPSNet++	c. Pan-Edge	64.8 / 58.2 / 69.5	57.8 / 45.2 / 66.4	54.5 / 39.8 / 65.0	52.6 / 36.2 / 64.5	57.4 / 45.1 / 66.4
VPSNet++	d. Tube-Disc	65.0 / 58.3 / 70.0	58.1 / 46.1 / 66.9	55.2 / 40.2 / 66.1	53.4 / 36.6 / 65.6	57.9 / 45.2 / 67.2
VPSNet++	e. a + d	65.4 / 58.2 / 70.7	58.8 / 46.2 / 67.9	55.3 / 40.3 / 66.2	53.4 / 36.6 / 65.6	58.2 / 45.3 / 67.6
VPSNet++	f. a + b + c + d	**65.6** / 58.3 / 70.9	**59.1** / 46.5 / 68.2	**55.6** / 40.5 / 66.5	**53.5** / 36.7 / 65.8	**58.4** / 45.5 / 67.9

TABLE V: **Video panoptic segmentation results on Cityscapes-VPS *val* set.** Each cell contains VPQ / VPQ^{Th} / VPQ^{St} scores. Note that while benchmarking our Cityscapes-VPS dataset, we further split our data into 400/50/50 (train/val/test) videos, which result in different performances to those reported in the CVPR 2020 version.

Model variant	Temporal window size				VPQ
	k = 0	k = 5	k = 10	k = 15	
Track	63.1 / 58.0 / 66.4	56.8 / 45.7 / 63.9	53.6 / 40.3 / 62.0	51.5 / 35.9 / 61.5	56.3 / 45.0 / 63.4
FuseTrack	64.2 / 59.0 / 67.7	57.9 / 46.5 / 65.1	54.8 / 41.1 / 63.4	52.6 / 36.5 / 62.9	57.4 / 45.8 / 64.8
VPSNet++	**65.7** / 59.9 / 69.4	**59.1** / 47.3 / 66.6	**55.5** / 41.8 / 64.3	**53.5** / 37.7 / 63.6	**58.5** / 46.7 / 66.0

TABLE VI: **Video panoptic segmentation results on Cityscapes-VPS *test* set.** Each cell contains VPQ / VPQ^{Th} / VPQ^{St} scores.

(a) Input frame (b) Initial optical flow (c) Refined feature flow (d) Predicted residual flow

Fig. 8: **Effect of flow refinement in Fuse head.** Example results of (b) initial optical flow computed from FlowNet2 [47] and (c) refined feature flow and (d) predicted residual flow after training for video panoptic segmentation. The structure of the scene is more pronounced in the refined feature flow.

class logit information not enough for differentiating these instances. On the other hand, IoU-Match is a simple yet strong candidate method for our task by leveraging spatial correlation to determine the instance labels, improving the Cls-Sort by +8.6% VPQ. The RoI feature matching (Feat-Match) performed by the Track head is based on the object appearance and is more robust to occlusions than IoU-Match. This improves VPQ by + 4.4%. Our final tracking method combines all three methods and achieves the best performance of 48.4% VPQ (*i.e.*, a further gain of +1.0% VPQ). This setting is used throughout the remaining experiments.

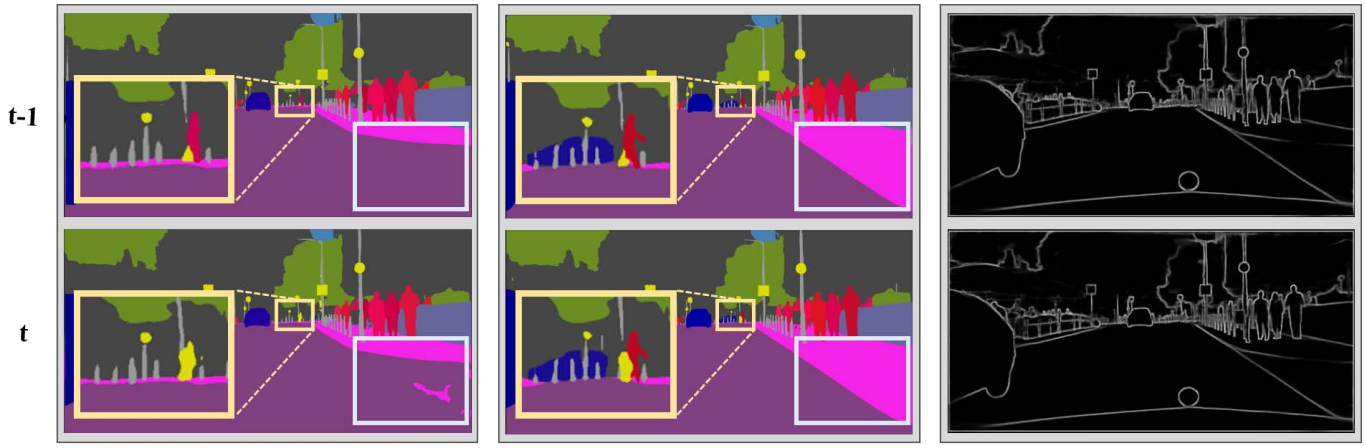
2) Bi-directional Fuse Head: We first confirm the effectiveness of pixel-level feature fusion by comparing the default Track and FuseTrack variants. Adding Fuse head leads to a gain of +1.8% VPQ on VIPER and +1.1% VPQ on Cityscapes-VPS dataset.

Our proposed Bi-directional Fuse head in VPSNet++ (denoted as 'a. Bi-Fuse' in the tables) further improves this by +1.1% VPQ on VIPER and +0.9% VPQ on Cityscapes-VPS dataset, demonstrating its benefit of making the reference and target features more compatible to each other, and helping feature similarity learning in Track head.

In Fig. 8, we visualize how the learnable feature flow is refined over the initial optical flow from the off-the-shelf FlowNet2. We can observe that the structure of the scene is more pronounced in the refined feature flow indicating that the pre-computed flow may not be optimal to find pixel correspondences for panoptic features.

3) Contrastive Track Head: The Track head of Yang *et al.* [30] learns object tracking by predicting multi-class classification where the classification labels are constructed by the previously detected object ids from the reference frame. We propose a noise contrastive estimation (NCE) loss function to model this *1 vs others* discrimination problem and enabled hard example mining. In Table. IV and Table. V, we compare the default Track head (FuseTrack), and our contrastive learning Track head (b. VPSNet++ with Contra-Track). Note the only difference between the to models is in the use of default vs Contrastive Track head. Contra-Track with the proposed NCE loss and hard negative mining allows focusing on difficult tracking scenarios, *e.g.*, occluded car and person (see Fig. 9), and leads to a gain of +0.7% VPQ on VIPER and +0.6% VPQ on Cityscapes-VPS.

4) Panoptic Edge Learning: The improvement by panoptic edge learning is denoted as 'c. Pan-Edge' in the tables. Note that the edge head for the auxiliary tasks is not used during inference, thus the use of auxiliary task does not burden



(a) Base VPSNet (b) VPSNet++ (c) Panoptic edge prediction

Fig. 9: **Result visualization of (a) VPSNet *vs* (b) VPSNet++ with ResNet-50 backbone on Cityscapes-VPS *val* set.** With the given consecutive time steps t-1 (top) and t (bottom), we show two video panoptic segmentation results, which are predicted from (a) the base VPSNet [14] and (b) VPSNet++ equipped with the Bi-directional Fuse head, contrastive learning based Track head and panoptic Edge prediction head. Although the Edge head is not necessary during inference, we visualize the predicted panoptic boundaries (c).

the model deployment. With the ability to estimate panoptic boundaries, our VPSNet++ can detect borders between different thing instances or between semantic classes, and thus can generate more accurate segmentation maps (see Fig. 9) Pan-Edge improves VPQ by +0.6% on VIPER and +0.4% on Cityscapes-VPS.

5) Panoptic Tube Id Discrimination Learning: The tube id discrimination learning alone outperforms FuseTrack by a healthy margin of +1.4% VPQ on VIPER and +0.9% VPQ on Cityscapes-VPS. Learning to discriminate different thing and stuff identities among the video panoptic features allows the well-clustered feature embedding, significantly contributing to the video panoptic segmentation.

6) VPSNet++ with All Improvements: Among the improvement methods of VPSNet++, the Bi-directional Fuse head (a. Bi-Fuse) and the Tube Id Discrimination learning (d. Tube-Disc) are the two strongest contributors. Combining these two improvements (e. a + d) achieves a significant gain of +1.7% VPQ on VIPER and +1.4% VPQ on Cityscapes-VPS over the FuseTrack model. Finally, Combining all the proposed improvements (f. a + b + c + d) improves this further by +0.4% VPQ on VIPER and +0.2% VPQ on Cityscapes-VPS. Compared to the FuseTrack baseline, our VPSNet++ improves this by +2.1% (50.5%) VPQ on VIPER, +1.4% (58.4%) VPQ on Cityscapes-VPS *val* set, and +1.1% (58.5%) VPQ on Cityscapes-VPS *test* set.

VII. Discussion and Conclusion

We present a new task named video panoptic segmentation (VPS). We contribute to the new benchmark by proposing 1) datasets 2) evaluation metric (VPQ) and 3) network architectures. We find several challenges still remaining for our new task. First, even the state-of-the-art video instance tracking algorithm [30] and our VPSNet and VPSNet++ suffer a considerable performance drop as the temporal length increases. In the context of video, possible improvements are expected to made on handling a large number of instances and resolving overlaps between these objects, *e.g.*, Fig. 7-(2nd row), by better modeling the temporal information [38], [27]. Second, our task is still challenging for *stuff* classes as well considering the fact that the window size of 15 frames represents only $0.5 \sim 1$ second in a video. The mutual exclusiveness between things and stuff class pixels could be further exploited to encourage both semantic segmentation and instance segmentation to regularize each other.

Other important research directions include improving the efficiency of an algorithm as in several video segmentation approaches [25], [26], [62], learning a Transformer-based video network [63], and extending to include depth estimation ability [64].

References

[1] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9404–9413.

[2] Q. Li, A. Arnab, and P. H. Torr, "Weakly-and semi-supervised panoptic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 102–118.

[3] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6399–6408.

[4] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang, "Attention-guided unified network for panoptic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7026–7035.

[5] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun, "Upsnet: A unified panoptic segmentation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8818–8826.

[6] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang, "An end-to-end network for panoptic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6172–6181.

[7] C.-Y. Fu, T. L. Berg, and A. C. Berg, "Imp: Instance mask projection for high accuracy semantic segmentation of things," in *ICCV*, 2019.

[8] J. Lazarow, K. Lee, and Z. Tu, "Learning instance occlusion for panoptic segmentation," *arXiv preprint arXiv:1906.05896*, 2019.

[9] J. Li, A. Raventos, A. Bhargava, T. Tagawa, and A. Gaidon, "Learning to fuse things and stuff," *arXiv preprint arXiv:1812.01192*, 2018.

[10] L. Porzi, S. R. Bulo, A. Colovic, and P. Kontschieder, "Seamless scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8277–8286.

[11] D. de Geus, P. Meletis, and G. Dubbelman, "Panoptic segmentation with a joint semantic and instance segmentation network," *arXiv preprint arXiv:1809.02110*, 2018.

[12] ——, "Single network panoptic segmentation for street scene understanding," *arXiv preprint arXiv:1902.02678*, 2019.

[13] T.-J. Yang, M. D. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, and L.-C. Chen, "Deeperlab: Single-shot image parser," *arXiv preprint arXiv:1902.05093*, 2019.

[14] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Video panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9859–9868.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.

[16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4990–4999.

[18] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for benchmarks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2213–2222.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[20] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 108–126.

[21] H. Wang, R. Luo, M. Maire, and G. Shakhnarovich, "Pixel consensus voting for panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9464–9473.

[22] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, and K. Huang, "Ssap: Single-shot instance segmentation with affinity pyramid," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 642–651.

[23] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 475–12 485.

[24] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[25] Y. Li, J. Shi, and D. Lin, "Low-latency video semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5997–6005.

[26] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell, "Clockwork convnets for video semantic segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 852–868.

[27] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2349–2358.

[28] D. Nilsson and C. Sminchisescu, "Semantic video segmentation by gated recurrent flow propagation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6819–6828.

[29] S. Jain, X. Wang, and J. E. Gonzalez, "Accel: A corrective fusion network for efficient semantic segmentation on video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8866–8875.

[30] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," *arXiv preprint arXiv:1905.04804*, 2019.

[31] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 221–230.

[32] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool, "Blazingly fast video object segmentation with pixel-wise metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1189–1198.

[33] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 686–695.

[34] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2663–2672.

[35] P. Tokmakov, K. Alahari, and C. Schmid, "Learning video object segmentation with visual memory," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4481–4490.

[36] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos, "Efficient video object segmentation via network modulation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6499–6507.

[37] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim, "Fast video object segmentation by reference-guided mask propagation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7376–7385.

[38] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," *arXiv preprint arXiv:1904.00607*, 2019.

[39] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3038–3046.

[40] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 408–417.

[41] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7942–7951.

[42] G. Bertasius and L. Torresani, "Classifying, segmenting, and tracking object instances in video with mask propagation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9739–9748.

[43] A. Athar, S. Mahadevan, A. Osep, L. Leal-Taixe, and B. Leibe, "Stem-seg: Spatio-temporal embeddings for instance segmentation in videos," in *European Conference on Computer Vision*. Springer, 2020, pp. 158–177.

[44] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 302–321, 2019.

[45] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 821–830.

[46] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.

[47] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[48] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "Edvr: Video restoration with enhanced deformable convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[49] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

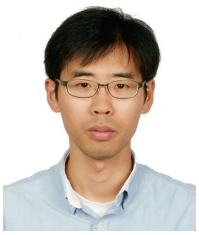
[50] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[51] J. Su, J. Li, Y. Zhang, C. Xia, and Y. Tian, "Selectivity or invariance: Boundary-aware salient object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3799–3808.

[52] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5229–5238.
[53] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.
[54] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
[55] J.-J. Hwang, S. X. Yu, J. Shi, M. D. Collins, T.-J. Yang, X. Zhang, and L.-C. Chen, "Segsort: Segmentation by discriminative sorting of segments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7334–7344.
[56] Z. Wu, A. A. Efros, and S. X. Yu, "Improving generalization via scalable neighborhood component analysis," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 685–701.
[57] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.
[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[59] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.
[60] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.
[61] K. Sofiiuk, O. Barinova, and A. Konushin, "Adaptis: Adaptive instance selection network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7355–7363.
[62] S. Woo, D. Kim, J.-Y. Lee, and I. S. Kweon, "Learning to associate every segment for video panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 2705–2714.
[63] D. Kim, J. Xie, H. Wang, S. Qiao, Q. Yu, H.-S. Kim, H. Adam, I. S. Kweon, and L.-C. Chen, "Tubeformer-deeplab: Video mask transformer," 2022.
[64] S. Qiao, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3997–4008.

Dahun Kim received his Ph.D., M.S. and B.S. degrees in Electrical Engineering from KAIST, Korea in 2022, 2018, and 2016, respectively. He received a Microsoft Research Asia Fellowship, Qualcomm Innovation Fellowship, and global doctoral fellowship funded by National Research Foundation (NRF). His research interests include object recognition, image/video processing, and computer vision. He is a member of IEEE.

Sanghyun Woo is a Ph.D student in Electrical Engineering department of Korea Advanced Institute of Science and Technology (KAIST), South Korea. He received his B.S and M.S degrees in Electrical Engineering from Seoul National University (SNU) and KAIST in 2017 and 2019 respectively. He received a Google Ph.D. Fellowship and Microsoft Research Asia Fellowship. His research interests include object recognition, image/video processing, and deep learning. He is a student member of the IEEE.



Joon-Young Lee is a Senior Research Scientist at Adobe Research. He received his Ph.D. and M.S. degrees in Electrical Engineering from KAIST, Korea in 2015 and 2009, respectively. He received the B.S. degree in Electrical and Electronic Engineering from Yonsei University, Korea in 2008. His research interests include deep learning and computer vision. He served as an Area Chair of ICCV 2019 and CVPR 2020.

In So Kweon is a professor in EE department of KAIST, South Korea. He received the BS and MS degrees in mechanical design and production engineering from Seoul National University, Korea, in 1981 and 1983, respectively, and the PhD degree in robotics from the Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, in 1990. He joined the Department of Automation and Design Engineering, KAIST, Seoul, Korea, in 1992, where he is now with the Department of Electrical Engineering. He was the program co-chair for the ACCV 07, general chair for ACCV 12, and program chair for ICCV 19. He is a member of the IEEE.