Vanishing-Point-Guided Video Semantic Segmentation of Driving Scenes

Diandian Guo[1]   Deng-Ping Fan[3,2*]   Tongyu Lu[1]   Christos Sakaridis[1]   Luc Van Gool[1]

[1] Computer Vision Lab, ETH Zürich   [2] DISSec, CS, Nankai University
[3] Nankai International Advanced Research Institute (SHENZHEN FUTIAN)

Abstract

*The estimation of implicit cross-frame correspondences and the high computational cost have long been major challenges in video semantic segmentation (VSS) for driving scenes. Prior works utilize keyframes, feature propagation, or cross-frame attention to address these issues. By contrast, we are the first to harness vanishing point (VP) priors for more effective segmentation. Intuitively, objects near VPs (i.e., away from the vehicle) are less discernible. Moreover, they tend to move radially away from the VP over time in the usual case of a forward-facing camera, a straight road, and linear forward motion of the vehicle. Our novel, efficient network for VSS, named **VPSeg**, incorporates two modules that utilize exactly this pair of static and dynamic VP priors: sparse-to-dense feature mining (DenseVP) and VP-guided motion fusion (MotionVP). MotionVP employs VP-guided motion estimation to establish explicit correspondences across frames and help attend to the most relevant features from neighboring frames, while DenseVP enhances weak dynamic features in distant regions around VPs. These modules operate within a context-detail framework, which separates contextual features from high-resolution local features at different input resolutions to reduce computational costs. Contextual and local features are integrated through contextualized motion attention (CMA) for the final prediction. Extensive experiments on two popular driving segmentation benchmarks, Cityscapes and ACDC, demonstrate that VPSeg outperforms previous SOTA methods, with only modest computational overhead. The resources are available at* https://github.com/RascalGdd/VPSeg.

1. Introduction

In the dawn of automated driving, a comprehensive understanding of the vehicle's surroundings becomes a must. Semantic segmentation, *i.e.*, the per-pixel classification of camera frames into a set of predefined classes, is a cen-
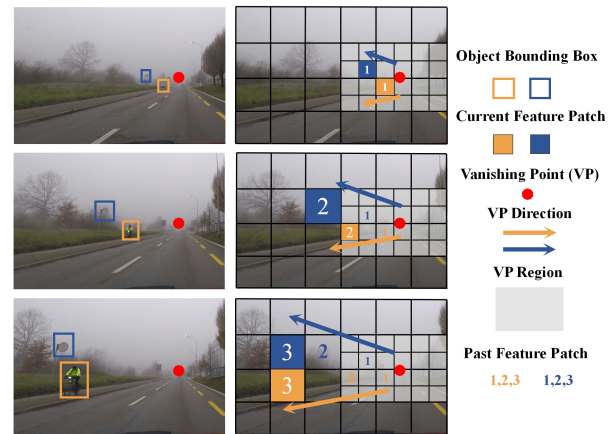

Figure 1. **Illustration of the intuition behind our proposed vanishing-point-guided motion estimation and scale-adaptive partition modules.** Targets move radially away from the vanishing point as time progresses in the video for the typical case of a forward-facing camera, a straight road and linear forward motion, which is depicted in this example. Moreover, the region around the vanishing point contains more distant objects, which appear at smaller scales.

tral task in this context. However, semantic segmentation in automated driving contexts presents unique challenges. The diversity of objects, their varying scales, potential occlusions, and the wide range of lighting and weather conditions create complex visual inputs that must be parsed in real time. Especially problematic are "hard-to-segment" objects [11] which are small, rare, or have an appearance that blends seamlessly into their background or into each other. These objects, such as distant traffic signs or badly lit pedestrians, are critically important to segment, given their impact on driving decisions.

As the dynamic context in consecutive frames provides clues to recognize these hard samples, researchers have tried to address this issue by exploiting temporal information through video semantic segmentation (VSS) [23, 33, 42, 43, 63]. However, processing multiple frames simultaneously requires significant computational resources. Moreover, existing VSS methods still struggle to establish correspondences in two main scenarios. First, for distant small objects, their relative motion over time tends to be

---
*Corresponding author: Deng-Ping Fan *(dengpfan@gmail.com)*.
Diandian Guo and Tongyu Lu share equal contributions to this project.

very subtle and can easily be overlooked. Second, in high-speed driving scenes, rapid changes in object positions and appearances can pose challenges to motion estimation. Typical methods assist VSS with optical flow [17, 49, 63], which not only fails for fast motions, but also introduces higher latency. Recent works have turned to local attention to leverage temporal information [23, 42]. However, while the coarser granularity in the attention mechanism enables a broader context, it risks neglecting fine motion characteristics. In addition, the non-dynamic feature tracking in local attention could easily miss the fast-moving features.

Inspired by the basics of perspective projection, we hypothesize that vanishing points (VPs) can provide useful priors for addressing the above issues in VSS of driving scenes. As seen in Fig. 1, the apparent motion of objects between consecutive frames in a video typically depends on the location of the VP, since static objects move radially away from the VP as time progresses in the usual case of a forward-facing camera, a straight road and linear forward motion. Therefore, this dynamic VP-related motion prior can serve as a constraint in motion estimation, leading to explicit cross-frame correspondences. Furthermore, the region of a frame which is located around the VP generally comprises distant parts of the driving scene, which consequently appear smaller. This static, intra-frame VP-related prior provides valuable context to quantitatively approximate and augment these crucial regions. Additionally, VP detection only requires the analysis of specific line segments or feature points in the frame, which significantly reduces the extra computational cost and does not slow down inference drastically. Thus, how to use these dynamic and static VP priors to guide VSS becomes a crucial question.

We address this question by proposing VPSeg, a VP-guided network for VSS. VPSeg leverages the above dynamic and static VP priors in two respective novel modules: the VP-guided motion fusion (MotionVP) and the sparse-to-dense feature mining (DenseVP). Specifically, MotionVP establishes explicit cross-frame correspondences through VP-guided motion estimation and it thus generates the dynamic context. On the other hand, DenseVP adopts a scale-adaptive partition strategy in the region around the VP, which we refer to as "VP region", to extract finer features for motions in this region, which are typically indistinct. Both MotionVP and DenseVP are implemented within a context-detail framework, where dynamic and local context are extracted from bilinearly downsampled low-resolution inputs. Subsequently, we fuse the local context with the dynamic context via contextualized motion attention (CMA) to obtain the detail attention map, which guides the integration of dynamic context with high-resolution features for the final prediction. Our contributions are summarized as:

- We propose MotionVP, a VP-guided motion estimation strategy for VSS, which yields explicit cross-frame cor-

respondences. MotionVP is particularly useful in high-speed scenarios in driving scenes, with large motion.
- We present DenseVP, a VP-guided scale-adaptive partition method for VSS, which extracts more fine-grained features for hard samples in the VP region.
- We design VPSeg, an efficient context-detail framework for VSS, which adaptively separates the contextual and detail-based features with different resolutions to reduce the computational cost on video frames.

2. Related Work

Semantic segmentation performs pixel-level labeling of an image with a set of object categories [34]. With the advent of deep learning, various segmentation networks have been proposed [3–5, 9, 13, 16, 18, 19, 24, 28, 54, 56, 59, 61, 62, 64] and harnessing richer *spatial* context has emerged as a primary theme for this task. In contrast, video semantic segmentation (VSS) involves performing semantic segmentation on consecutive video frames and further exploits the *temporal* context. Existing VSS methods can roughly be categorized into efficient VSS and high-performance VSS. While the former category compromises accuracy to speed up segmentation by reusing the features, the latter strives to enhance current frame segmentation using expensive per-frame networks. Our method falls into the second category. By guiding VSS via vanishing point priors, we achieve a better tradeoff between performance and model complexity.

2.1. Efficient VSS

Efficient VSS aims to reduce the computational cost and improve the segmentation efficiency [15, 17, 22, 26, 32, 40, 53, 63]. The most typical kind of efficient VSS approaches is the keyframe method, where the model applies expensive feature extraction and segmentation networks only on keyframes, while non-keyframe features are fine-tuned from keyframe features to reduce computation [17, 22, 26, 32, 53, 63]. Among these works, Accel [17] uses optical flow for feature warping [49]. DFF [63] propagates deep feature maps through a flow field. DVSNet [53] embraces an adaptive keyframe and region scheduling policy. In addition, various other works [15, 40] also adopt the feature propagation method, where the features in preceding frames are reused to accelerate computation. Although these methods have improved the segmentation efficiency through feature sharing and propagation, their usage of proxy features often leads to inaccurate results.

2.2. High-Performance VSS

High-performance VSS focuses on enhancing accuracy by leveraging the temporal continuity of input videos [23, 29, 33, 36, 42, 43]. Unlike efficient VSS approaches, these methods employ per-frame networks, utilizing a full, costly segmentation network for each frame and enhancing the
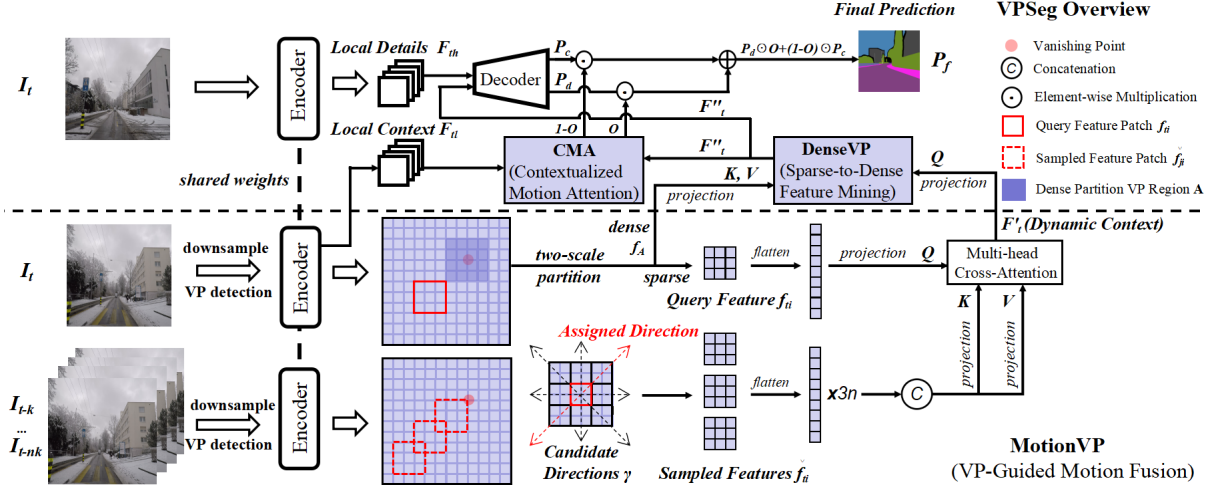
Figure 2. Overview of our VPSeg network. In the MotionVP module (bottom part), video frames are downsampled to extract context features, which go through cross-attention to capture dynamic context F'_t. F'_t is further augmented by DenseVP to mine finer features f_A in the VP region through a two-scale partition strategy. In the upper part, we obtain local context F_{tl} and local details F_{th} from downsampled and high-resolution target frames I_t, respectively. In CMA, augmented dynamic context F''_t interacts with local context F_{tl} to generate the detail attention map O, guiding its merging with high-resolution local details F_{th} for the final prediction P_f. Zoomed in for best view.

current frame segmentation by mining temporal correlations from video frames. For example, Liu *et al.* [29] considered the temporal consistency among frames as extra constraints by using knowledge distillation for more robust VSS. Sun *et al.* [43] estimated cross-frame affinities to enhance temporal information aggregation. Another trend in recent works [36, 42, 43] is the use of attention mechanisms [46], where the model dynamically focuses on specific parts of a video sequence to better exploit the temporal context. However, the high computational demands of these methods typically limit their application to low-resolution inputs and render them impractical when aiming for high-resolution segmentations of video frames.

2.3. VP Detection

The vanishing point (VP) is a geometric quantity in perspective projection, which constitutes the point of apparent convergence of parallel 3D lines in the 2D image. VPs are involved in many applications, such as camera calibration [1], lane departure warning [55], and mapping [27]. In contemporary research, numerous VP detection approaches have been developed. In general terms, these methods can be divided into two categories: traditional, hand-crafted methods and deep learning-based methods.

Traditional methods mainly include texture detection methods [35, 41, 65] and edge detection methods [20, 44]. Texture detection methods search for the dominant direction of textures in images and then vote for the VP location. They require heavy computation and cannot run in real time. In structured road scenes, Hough-transform-based [31] edge detection methods [10, 12] are more commonly used. They extract lines and transform them into Hough space to allow voting for the candidate VP. Deep

learning-based methods [2, 52, 58] primarily use CNNs [21] to directly predict the VP location from raw image pixels. However, a lack of dedicated automated driving datasets, combined with longer inference times, hampers their practical applicability. In this work, we adopt the Hough-transform-based edge detection method [10, 31] to strike a balance between accuracy and inference speed.

3. VPSeg: VP-Guided Network for VSS

Motivation. Structural cues such as depth maps [7], layouts [45], and textures [50] have proven essential for scene understanding. However, their application to VSS has been scarcely researched. On the one hand, such models as for depth estimation have unstable performance on distant regions. On the other hand, most deep learning-based methods are time-consuming and cumbersome, making them impractical to combine with other tasks.

On the contrary, VP detection is swift and robust, particularly through the application of edge detection techniques across a wide range of structured driving scenes. Furthermore, motion information based on the locations of a VP in the frames of the input video can serve to capture cross-frame correspondences, given that the apparent motion of pixels across frames is often aligned with their respective offsets from this VP (cf. Fig. 1). Consequently, how to exploit such VP-guided motion priors constitutes an unexplored and highly relevant question for VSS. Notably, contemporary high-performance VSS methods are computationally demanding. Recent endeavors in this field either necessitate expensive hardware or are limited to low-resolution datasets in terms of applicability (*e.g.*, as seen in [33, 42, 43]). Given these findings, we pose another ques-

tion: how to utilize temporal information in video frames more efficiently while maintaining high accuracy?

We found a positive answer for these two questions. Through VP-guided motion fusion (MotionVP), we establish explicit cross-frame correspondences, extracting relevant dynamic features from adjacent frames. Sparse-to-dense feature mining (DenseVP) adopts an adaptive partition of the input frame to mine finer features for subtle motions in the VP region. Additionally, our context-detail framework separates the extraction of context from detail-based features with different input resolutions. We integrate high-resolution local predictions with downsampled contextual predictions using contextualized motion attention (CMA) to reduce computation. Fig. 2 illustrates the architecture of our novel VPSeg network, which incorporates the above newly proposed modules.

3.1. MotionVP: VP-Guided Motion Fusion

VSS poses the challenge of establishing cross-frame feature correspondences for the same object in frame sequences. We observe that in typical automated driving scenarios, the relative movements of static and dynamic objects normally follow the lane markings (see Fig. 1). Typically, the driving direction coincides with the direction of the VP, which implies that most objects move radially away from the VP as time progresses in the video.

To utilize this VP dynamic prior for motion estimation, we initialize four candidate orientations, $\gamma = n \cdot \frac{\pi}{4}, n = 0, 1, 2, 3$. Their corresponding vector representations can be denoted as $\mathcal{V} = \{(+1, 0), (+1, +1), (0, +1), (-1, +1)\}$. Assume we have a training data point containing $n + 1$ video frames $\mathcal{I} = \{I_{t-nk}, ..., I_{t-2k}, I_{t-k}, I_t\}$ with corresponding timestamps $\mathcal{T} = \{t - nk, ..., t - 2k, t - k, t\}$, where $t - k$ is k frames earlier than t and k is the frame sampling interval. We employ the specified n previous frames to enhance the semantic segmentation of the target frame I_t. First, we extract context feature maps $\mathcal{F} = \{F_{t-nk}, ..., F_{t-2k}, F_{t-k}, F_t\}$ from the bilinearly downsampled video frames $\mathcal{I}$ with a pre-trained transformer encoder, where $F \in \mathbb{R}^{c \times h \times w}$ and c, h, w represent channels, height and width, respectively. Then we partition the feature map into feature blocks of size $s \times s$. For the i-th feature patch index (x_i, y_i) in patch index set $\mathcal{D} = \{(x_i, y_i) \in \mathbb{N}^2 | x_i < \frac{w}{s}, y_i < \frac{h}{s}\}$, the corresponding feature patch in frame j is denoted as $f_{ji} \in \mathbb{R}^{c \times s^2}$. For each patch (x_i, y_i), we consider the vector pointing from the patch center to the VP as its motion direction:

$$(\Delta x_{ji}, \Delta y_{ji}) = (\hat{x}_j - x_i, \hat{y}_j - y_i), \tag{1}$$

where $(\hat{x}_j, \hat{y}_j)$ is the estimated patch-level VP position in frame j. Different from $(x_i, y_i) \in \mathcal{D}$, we have $(\hat{x}_j, \hat{y}_j) \in [0, \frac{w}{s} - 1] \times [0, \frac{h}{s} - 1]$. Subsequently, we mark the candidate direction that most closely matches the motion direction of the patch as its "assigned direction". The assigned direction

(u_{ji}, v_{ji}) for patch (x_i, y_i) in frame j can be computed as

$$(u_{ji}, v_{ji}) = \underset{(u,v) \in \mathcal{V}}{\operatorname{argmin}} \operatorname{dist}((u, v), (\Delta x_{ji}, \Delta y_{ji})), \tag{2}$$

where $\operatorname{dist}(\cdot)$ quantitatively computes the difference between the motion direction and each candidate direction:

$$\operatorname{dist}((u, v), (\Delta x, \Delta y)) = |\alpha(u, v) - \alpha(\Delta x, \Delta y)|, \tag{3}$$

where $\alpha(u, v) = \operatorname{NumPy.arctan2}(v, u)$. Then we sample adjacent patches forward and backward along the assigned direction to capture bidirectional correspondences. As the frame interval $t - j$ increases, the sampling distance $(u_{ji}, v_{ji})'$ also increases linearly with the sampling coefficient Δd, where a larger Δd suits higher driving speeds and larger frame sampling intervals k:

$$(u_{ji}, v_{ji})' = \frac{t - j}{k} \times \Delta d \times (u_{ji}, v_{ji}). \tag{4}$$

Now, we have sampled features $\check{f}_{ji} \in \mathbb{R}^{c \times 3s^2}$ from the concatenation of forward, backward, and local sampled patches $(\check{x}_{ji}^f, \check{y}_{ji}^f)$, $(\check{x}_{ji}^b, \check{y}_{ji}^b)$, and $(\check{x}_{ji}^l, \check{y}_{ji}^l)$ from frame j:

$$\begin{aligned} (\check{x}_{ji}^f, \check{y}_{ji}^f) &= (x_i, y_i) + (u_{ji}, v_{ji})', \\ (\check{x}_{ji}^b, \check{y}_{ji}^b) &= (x_i, y_i) - (u_{ji}, v_{ji})', \\ (\check{x}_{ji}^l, \check{y}_{ji}^l) &= (x_i, y_i). \end{aligned} \tag{5}$$

Given the local features $f_{ti} \in \mathbb{R}^{c \times s^2}$ for the i-th patch from the current frame I_t and sampled features from neighboring frames $\mathcal{S} = \left\{ \check{f}_{ji} \in \mathbb{R}^{c \times 3s^2}, j \in \mathcal{T} \text{ and } j \neq t \right\}$, we use f_{ti} as queries, $\mathcal{S}$ as keys and values for interaction:

$$Q_i = \mathrm{W}_q(f_{ti}), \quad K_i = \mathrm{W}_k(\mathrm{C}(\mathcal{S})), \quad V_i = \mathrm{W}_v(\mathrm{C}(\mathcal{S})), \tag{6}$$

where $\mathrm{W}(\cdot)$ and $\mathrm{C}(\cdot)$ represent fully connected layers and concatenation, respectively. Next, we use cross-attention $\mathrm{CA}(\cdot)$ [46] to compute patch-level dynamic features $f_{ti}' \in \mathbb{R}^{c \times s^2}$ for patch (x_i, y_i) of frame t:

$$f_{ti}' = \mathrm{CA}(Q_i, K_i, V_i). \tag{7}$$

The patch-level dynamic features of all patches are then simply tiled together to reconstruct the complete frame-level dynamic features $F_t' \in \mathbb{R}^{c \times h \times w}$ for frame t.

In summary, the VP-guided motion estimation establishes explicit feature correspondences by imposing constraints on motion estimation. This fast and simple algorithm also suits well high-speed scenarios with gradually increasing sampling distances $(u_{ji}, v_{ji})'$. It should be noted that low-resolution frames are utilized in our MotionVP module to prioritize context features over high-resolution details for more comprehensive dynamic contextual understanding. The dynamic context is afterwards fused with high-resolution details through contextualized motion attention (cf. Sec. 3.3) to obtain the final semantic prediction.

3.2. DenseVP: Sparse-to-Dense Feature Mining

In the vicinity of the VP, objects typically appear tiny and exhibit very small motions across frames. Therefore, we
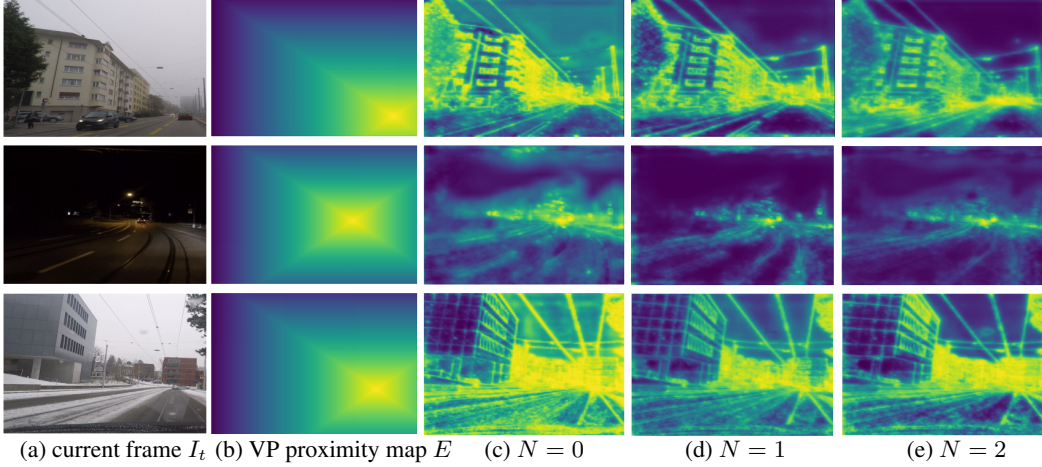
(a) current frame I_t (b) VP proximity map E (c) $N = 0$ (d) $N = 1$ (e) $N = 2$

Figure 3. **Visualization of detail attention maps** O **with** N **motion attention layers in CMA.** As N increases, the detail attention map interacts more heavily with the dynamic features, and the weights gradually decrease in closer parts of the scene or on simple semantic categories. The highlighted distant regions near the VP suggest that the final predictions P_f are primarily based on the detail-based predictions P_d and not on P_c for these regions. The VP proximity map serves as a positional prior and assists the model in pinpointing the locations of these distant regions.

propose a scale-adaptive VP-guided feature enhancement module called sparse-to-dense feature mining (DenseVP). In simple terms, we combine the dense and sparse partition to mine finer features for motions near the VP which are typically indistinct, while a sparser and more computationally efficient patch partition approach is adopted in the rest of the frame. Due to the instability in VP detection, we utilize the VP region, *i.e.*, the region that surrounds the VP, as a coarser and more robust alternative. Specifically, we first determine the VP region based on the position of the VP. Following the notation of Sec. 3.1, we have $\frac{h}{s} \times \frac{w}{s}$ feature patches, index (x_i, y_i) for the i-th feature patch and estimated patch-level VP position $(\hat{x}_j, \hat{y}_j)$ in frame j. The VP patch (x'_j, y'_j) represents the closest feature patch to VP and can be formulated by:

$$(x'_j, y'_j) = \underset{(x,y) \in \mathcal{D}}{\operatorname{argmin}}[(x - \hat{x}_j)^2 + (y - \hat{y}_j)^2]. \quad (8)$$

The VP region $\mathcal{A} \subset \mathcal{D}$ is a rectangular region near the VP patch containing $(2a + 1) \times (2b + 1)$ patches arranged in a sparse grid, with $a, b \in \mathbb{N}$:

$$\mathcal{A} = \left\{ (m, n) | (m - x'_j)^2 \leq a^2 , (n - y'_j)^2 \leq b^2 \right\}. \quad (9)$$

Then, an overlapping dense grid partition is applied in the VP region. More precisely, we adopt a stride of $\lceil s/2 \rceil$ for this dense partition, resulting in m overlapping patches, where $m = (\lfloor \frac{2as}{\lceil s/2 \rceil} \rfloor + 1)(\lfloor \frac{2bs}{\lceil s/2 \rceil} \rfloor + 1)$. Subsequently, the dynamic context F'_t is used as queries, while the features extracted densely from the VP region, $f_{\mathcal{A}} \in \mathbb{R}^{c \times ms^2}$, are used as keys and values:

$$Q_{\mathcal{A}} = W_q(F'_t), \quad K_{\mathcal{A}} = W_k(f_{\mathcal{A}}), \quad V_{\mathcal{A}} = W_v(f_{\mathcal{A}}), \quad (10)$$

where $W(\cdot)$ represents fully connected layers. Cross-attention operations $CA(\cdot)$ [46] are then employed to augment the dynamic context F'_t with dense features $f_{\mathcal{A}}$ for finer representations of motions in the VP region:

$$F''_t = CA(Q_{\mathcal{A}}, K_{\mathcal{A}}, V_{\mathcal{A}}), \quad F''_t \in \mathbb{R}^{c \times h \times w}. \quad (11)$$

The described two-scale feature partition enhances the dynamic context in the VP region, which typically contains distant objects, by leveraging static VP positional information. Thus, DenseVP exploits a static prior related to the position of the VP in a frame, while MotionVP exploits a dynamic VP prior related to apparent motion of objects depending on their relative position with respect to the VP.

3.3. CMA: Contextualized Motion Attention

Given the augmented dynamic context $F''_t \in \mathbb{R}^{c \times h \times w}$, we aim to integrate it with the high-resolution details for the final prediction. For the target frame I_t, the low-resolution and high-resolution inputs are denoted as I_{tl} and I_{th}, respectively. The corresponding local context and detail-based local features are $F_{tl} \in \mathbb{R}^{c \times h \times w}$ and $F_{th} \in \mathbb{R}^{c \times h \times w}$. First, we randomly initialize our learnable queries $Q \in \mathbb{R}^{c \times K}$ (K is the number of classes) and contextualize them with F_{tl} through VP-aware cross-attention $CA_E(\cdot)$:

$$Q_c = CA_E(W_q(Q), W_k(F_{tl}), W_v(F_{tl})), \quad (12)$$

where $W(\cdot)$ denotes fully connected layers and $Q_c \in \mathbb{R}^{c \times K}$ are the contextualized queries. $CA_E(\cdot)$ is given by

$$CA_E(Q, K, V) = \operatorname{Softmax}(\frac{QK^T}{\sqrt{c}} + E)V + Q. \quad (13)$$

The term E refers to our VP proximity map embedding. It is a VP-centered pseudo-depth map, where the depth of pixel (x, y) is $1 - \Delta D$, $\Delta D \propto \max\{\frac{|y - \hat{y}^p_j|}{h}, \frac{|x - \hat{x}^p_j|}{w}\}$ and $(\hat{x}^p_j, \hat{y}^p_j)$ is the VP pixel coordinate. Then, we perform motion attention to merge local and dynamic context:

$$F_m = CA_E(W_q(Q_c), W_k(F''_t), W_v(F''_t)), \quad (14)$$

where the merged context $F_m \in \mathbb{R}^{c \times K}$. Thus, the detail attention map $O \in \mathbb{R}^{K \times h \times w}$ is computed as

$$O = F_m^T F_{tl}. \tag{15}$$

The final prediction P_f is the weighted sum of the context prediction P_c and the detail-based prediction P_d under the guidance of the detail attention map O:

$$P_f = (1 - O) \odot P_c + O \odot P_d, \tag{16}$$

where $\odot$ denotes element-wise multiplication, and P_c and P_d are generated from F_{tl} and F_{th} through the DAFormer [14] decoder. For every different semantic category and every distinct position, O learns to weight the high-resolution details and dynamic context differently. As shown in Fig. 3, the weights for P_d are generally higher in distant regions, indicating that the final prediction P_f in these zones are primarily determined by the detail-based prediction P_d. By contrast, in closer regions and simple semantic categories, the final results mostly rely on the context prediction P_c. Our VP proximity map serves as a positional prior embedding, guiding the detail attention map O to prioritize those deeper regions. The loss function of VPSeg is denoted as

$$\mathcal{L}_{\text{total}} = (1 - \lambda_d)\mathcal{L}_{\text{CE}}(P_f, G) + \lambda_d \mathcal{L}_{\text{CE}}(P_d, G), \tag{17}$$

where G and λ_d represent ground truth and the loss coefficient for detail-based prediction. Here we compute the final loss as the weighted sum of the cross-entropy loss from fused and detail-based prediction, which is beneficial for improving the high-resolution details while optimizing the feature fusion.

4. Experiments
4.1. Experimental Setup
Implementation details. Our end-to-end model is trained for 160k iterations using the AdamW [30] optimizer. We set the batch size to 4, with an initial learning rate of 2×10^{-4}, and the weight of the detail-based prediction loss to $\lambda_d = 0.1$. Following the settings of SegFormer [51], we employ the MiT [51] pre-trained on ImageNet [38] as the backbone. For generality, only previous frames are utilized to assist segmentation for each current frame. The frame sampling interval is set to $k = 3$ and the number of neighboring frames in each batch is 3. We adopt random resizing, flipping, and cropping for data augmentation. 1024×2048 resolution inputs are employed in VPSeg and the training time for ACDC/Cityscapes is 28/32 hours with MiT-B3 backbone. For context features and VP detection, inputs are bilinearly downsampled with a ratio of 0.5.

For VP estimation, the thresholds of the Canny edge filter are set to 50 and 150. The sampling coefficient Δd in MotionVP is 1. For DenseVP, we define the VP region as 3×3 sparse patches ($a = 1, b = 1$). During inference, we carry out a single-scale sliding window test. All experiments are conducted on 4 NVIDIA 3090 GPUs.

Datasets. Our experiments are primarily performed on two automated driving datasets: ACDC [39] and Cityscapes [8]. ACDC consists of a large collection of images ($1,600$ clips for training and 406 clips for validation) that are evenly distributed across four common adverse conditions: fog, nighttime, rain, and snow. More importantly, ACDC's annotation strategy involves a two-step process that creates a binary "invalid mask" to highlight ambiguous image regions. Initially, a semantic label is manually drafted and then refined using adverse condition videos to finalize the annotation, allowing a valid quantitative assessment of segmentation in uncertain areas. We also perform experiments on the Cityscapes video dataset, which includes 3,475 clips from 21 cities in its training and validation splits.

Evaluation metrics. The experiment results are evaluated on three metrics: mIoU, iIoU [8], and IA-IoU. As mIoU tends to downweigh small instances within the same class, we also employ the instance-level intersection-over-union (iIoU) for more informative evaluation with respect to performance on tiny objects. iIoU is defined as

$$i\text{IoU} = \frac{i\text{TP}}{i\text{TP} + i\text{FN} + \text{FP}}, \tag{18}$$

where TP, FN and FP represent true positive, false negative and false positive, respectively. The iIoU assigns higher weights i to pixels in smaller instances.

For ACDC [39], we propose a new metric IA-IoU (invalid-area intersection-over-union). As discussed above, the invalid masks of ACDC are particularly informative as they reveal regions of ambiguity or what might be termed as uncertain regions within the frame. Therefore, we test mIoU explicitly on these invalid regions as IA-IoU. Specifically, suppose we have K semantic classes and n_{max} validation images. With the corresponding invalid mask M_n for the n-th image, we have:

$$\hat{P}_{nz} = P_{nz} \cap M_n, \tag{19}$$

where P_{nz} is the model prediction of the z-th class in the n-th image. Similarly, for the ground truth:

$$\hat{G}_{nz} = G_{nz} \cap M_n, \tag{20}$$

where G_{nz} is the ground truth of the z-th class in the n-th image. The IA-IoU of class z can be calculated as

$$\text{IA-IoU}_z = \frac{\sum_{n=0}^{n_{max}} |\hat{P}_{nz} \cap \hat{G}_{nz}|}{\sum_{n=0}^{n_{max}} |\hat{P}_{nz} \cup \hat{G}_{nz}|}. \tag{21}$$

The mean IA-IoU (mIA-IoU) is the average of IA-IoU for each class. This specialized metric allows an evaluation targeted specifically to the regions which are marked as uncertain and ambiguous in the ground truth.

4.2. Comparison with the State of the Art
In Tab. 1, we compare the performance of VPSeg to state-of-the-art methods on ACDC and Cityscapes. Qualitative results are shown in Fig. 4. We observe that our method

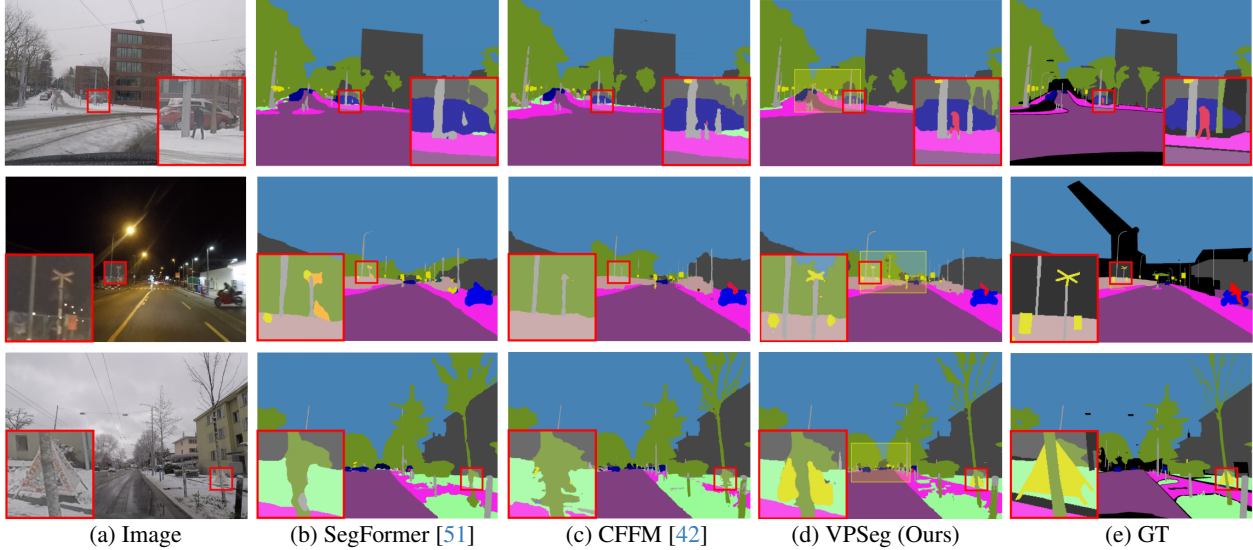| | (a) Image | (b) SegFormer [51] | (c) CFFM [42] | (d) VPSeg (Ours) | (e) GT |

Figure 4. **Qualitative comparison on ACDC.** The yellow box represents the densely partitioned VP region. Our model produces more accurate results for both distant tiny hard samples near the VP and occluded fast-moving close targets.

| Methods | Backbones | Params (M)↓ | mIoU↑ | | miIoU↑ | | mIA-IoU↑ | VSS |
			ACDC	Cityscapes	ACDC	Cityscapes	ACDC	
DeepLabv3+[6]	ResNet-101	62.7	72.79	79.09	43.14	56.89	36.32	✗
PSPNet [60]	ResNet-101	68.0	72.26	78.34	40.95	56.78	37.29	✗
OCRNet [57]	ResNet-101	55.6	70.39	80.09	43.36	59.55	33.38	✗
SeaFormer [47]	SeaFormer-L	14.0	70.09	77.70	40.41	56.96	33.26	✗
SegFormer [51]	MiT-B1	**13.7**	70.25	78.56	41.14	58.25	34.05	✗
SegFormer [51]	MiT-B3	44.6	75.38	81.32	46.51	60.01	38.42	✗
Video K-Net [25]	Swin-B	104.6	69.03	76.62	39.33	56.02	31.67	✓
ETC [29]	ResNet-101	68.1	71.45	79.50	42.28	58.35	36.71	✓
TCB [33]	ResNet-101	72.5	70.56	78.70	41.76	57.84	35.96	✓
NetWarp [49]	PSPNet	90.6	73.71	80.60	45.59	59.63	36.58	✓
CFFM [42]	MiT-B3	49.6	75.47	81.44	47.31	60.09	37.88	✓
MRCFA [43]	MiT-B3	48.2	75.63	81.31	46.28	60.56	38.81	✓
VPSeg (Ours)	MiT-B1	14.9	72.86	79.56	43.42	59.53	37.96	✓
VPSeg (Ours)	MiT-B3	46.8	**77.48**	**82.46**	**49.42**	**61.79**	**41.48**	✓

Table 1. **Comparison with state-of-the-art methods on the ACDC [39] and Cityscapes [8] validation sets.** Our model outperforms the compared methods in mIoU, miIoU [8] and mIA-IoU.

captures indistinct and fast motions more accurately, leading to more robust segmentation.

On ACDC, VPSeg improves the mIoU by 2.10% and 1.85% compared to the baseline model SegFormer [51] and the SOTA VSS model MRCFA [43], respectively. More notable is our performance on mIA-IoU. For this metric that places more emphasis on indistinct regions, our model obtains even higher advancement, improving by 3.06% and 2.67% upon SegFormer and MRCFA, respectively. In Tab. 2, we display the IA-IoU performance of different models for selected categories. For hard categories such as rider (+8.5%) and traffic sign (+3.4%), VPSeg delivers significant improvements. At the same time, our method matches the performance of competing methods on easier categories with larger segments, including sky (+0.2%) and bus (+0.1%). We also provide results on miIoU in Tab. 1, where VPSeg performs favorably against the SOTA method CFFM [42] (+2.11%), indicating our model is better at handling distant, hard-to-segment small objects.

Besides ACDC, VPSeg also sets the new state of the art

on Cityscapes. Compared to the baseline model SegFormer and the VSS model CFFM, our mIoU is boosted by 1.14% and 1.02%, respectively. As for miIoU, VPSeg improves performance even more (+1.78%, +1.70%), evidencing its robustness in segmenting tiny objects. Both for MiT-B1 [51] and MiT-B3 [51] backbones, VPSeg clearly outperforms the corresponding baselines, showing that the proposed modules are general.

As is shown in Tab. 3, the resource efficiency of our method is also noteworthy. In contrast to SegFormer with MiT-B3 backbone, VPSeg adds only 2.2M extra parameters, which correspond to 4.9% of the total parameters of SegFormer (44.6M). Relative to other high-performance VSS models such as Video K-Net [25] and MRCFA, our method reaches superior computational efficiency with 1124.7 GFLOPs. Although limited by the VP inference speed (see Tab. 5), the FPS of VPSeg is still comparable. Overall, VPSeg delivers SOTA segmentation performance with only limited additional computational cost.

	person	rider	car	bus	VG	bicycle	TS	sky	road
DeepLabv3+ [6]	48.2	18.2	41.3	40.1	69.6	26.1	35.0	84.7	67.7
SegFormer [51]	47.9	19.0	44.6	45.9	71.1	35.4	32.6	85.4	**74.8**
SeaFormer [47]	41.2	10.6	36.2	41.5	70.1	29.7	29.4	84.5	62.9
ETC [29]	45.8	15.5	41.1	33.5	**71.6**	29.6	34.6	84.5	70.6
CFFM [42]	48.6	21.4	**46.6**	46.5	70.8	32.8	31.9	84.6	74.2
MRCFA [43]	47.8	24.3	46.1	46.1	71.3	35.7	34.1	85.1	74.7
VPSeg (Ours)	**51.8**	**32.8**	46.3	**46.6**	71.5	**39.3**	**38.4**	**85.6**	74.5

Table 2. **Per-class IA-IoU of methods from Tab. 1 on ACDC.** "VG": vegetation, "TS": traffic sign. SegFormer and VPSeg use an MiT-B3 [51] backbone.

Methods	Backbones	Params (M)↓	mIoU↑	GFLOPs↓	FPS↑
Video K-Net [25]	Swin-B	104.6	69.03	1430.0	-
TMANet [48]	ResNet-101	54.2	71.62	1385.9	2.4
ETC [29]	ResNet-101	68.1	71.45	-	1.2
TCB [33]	ResNet-101	72.5	70.56	-	1.9
MRCFA [43]	MiT-B3	48.2	75.63	1436.4	**4.4**
CFFM [42]	MiT-B3	49.6	75.47	1534.8	3.8
VPSeg (Ours)	MiT-B3	**46.8**	**77.48**	**1124.7**	3.4

Table 3. Comparison of FPS and GFLOPs of high-performance VSS methods on ACDC.

VP region size	mIoU (A.)↑	mIA-IoU (A.)↑	mIoU (C.)↑
No DenseVP	77.15	41.02	82.19
$a=0, b=0$	77.28	40.87	82.34
$a=1, b=1$	**77.48**	**41.48**	82.46
$a=2, b=2$	77.36	41.33	82.37
$a=3, b=3$	77.41	41.32	**82.50**

Table 4. Ablation study on the VP region size on ACDC (A.) and Cityscapes (C.) with MiT-B3 backbone.

Embeddings	mIoU (A.)↑	mIA-IoU (A.)↑	mIoU (C.)↑	FPS↑
No embedding	77.02	40.91	82.16	**4.2**
Depth map	77.19	40.78	82.23	1.8
VP proximity map	**77.48**	**41.48**	**82.46**	3.4
Depth map + VP	77.46	41.37	82.39	1.6

Table 5. Ablation study of different positional embeddings on ACDC (A.) and Cityscapes (C.) with MiT-B3 backbone.

4.3. Ablation Studies

Benefit of DenseVP. In Tab. 4, we conduct ablation experiments on the sparse-to-dense feature mining (DenseVP) module under different settings. As the VP region expands, the performance of the model gradually improves and peaks when 3×3 ($a=1, b=1$) feature patches are used. IA-IoU is more significantly influenced by DenseVP than mIoU, indicating that the scale-adaptive partition in DenseVP contributes more to the enhancement of uncertain regions.

Effect of different positional embeddings. Given that VP proximity maps are constructed as pseudo depth maps, we attempt to replace the VP proximity embedding in CMA with depth map embeddings. The experimental results are presented in Tab. 5, where the depth maps are generated by MiDaS V3-Hybrid [37]. As the depth estimation has unstable performances at distant regions, the depth map embedding yields limited improvement in mIoU and leads to worse performance on IA-IoU. In addition, the longer inference time of MiDaS results in a significant drop in the FPS. The VP proximity embedding alone reaches the best mIoU and mIA-IoU with a medium inference speed. Ex-

Methods / k	3	5	7	9
Segformer [51]	75.38	75.38	75.38	75.38
MRCFA [43]	75.63	75.44	75.32	75.30
CFFM [42]	75.47	75.35	75.22	75.07
VPSeg (Ours)	**77.48**	**77.52**	**77.39**	**77.43**

Table 6. Ablation study of the frame sampling interval k (default 3) on ACDC with MiT-B3 backbone.

N	mIoU (A.)↑	mIA-IoU (A.)↑	mIoU (C.)↑	Params (M)↓
0	76.65	40.33	82.12	**45.9**
1	77.21	41.23	82.33	46.4
2	77.48	**41.48**	**82.46**	46.8
3	**77.51**	41.39	82.44	47.3

Table 7. Ablation study on the number of motion attention layers in CMA with MiT-B3 backbone.

periments incorporating both depth map and VP proximity embeddings have comparable results, but still limited FPS.

Effect of the frame sampling interval k. We study the impact of the frame sampling interval k in Tab. 6. Notably, VPSeg obtains even higher mIoU when k increases from 3 to 5. For larger k, VPSeg also exhibits more robustness against other methods whose mIoU drops drastically. The possible reason is that the axial motion searching along the VP direction covers a wider range of motions, which is especially relevant for high-speed scenarios.

Influence of the number of motion attention layers N. Tab. 7 shows the performance of VPSeg with respect to the number of motion attention layers N in CMA. As N increases, the model performance significantly improves, proving the effectiveness of our dynamic feature fusion. We observe diminishing returns in performance for $N \geq 2$. For VPSeg with MiT-B3 backbone, we choose $N = 2$ for the best tradeoff between performance and model complexity.

5. Conclusion

The establishment of cross-frame correspondences and the reduction of computational cost are two pressing issues in VSS of driving scenes. We have proposed the novel VPSeg network to address these issues by exploiting dynamic and static VP priors through the novel MotionVP and DenseVP modules. The former module establishes explicit correspondences via a VP-guided motion estimation strategy, while the latter augments fine dynamic features in the region around the estimated VP through a scale-adaptive partition method. In the context-detail framework of VPSeg, the downsampled contextual features and high-resolution local details are separated and adaptively fused through our motion-aware CMA attention module. VPSeg has achieved SOTA performance for VSS on two widely used driving datasets at a reasonable computational cost.

6. Acknowledgement

References

[1] Michel Antunes, João P. Barreto, Djamila Aouada, and Björn Ottersten. Unsupervised vanishing point detection and camera calibration from a single manhattan image with radial distortion. In *IEEE CVPR*, 2017. 3

[2] Chin-Kai Chang, Jiaping Zhao, and Laurent Itti. Deepvp: Deep learning for vanishing point detection on one million street view images. In *IEEE ICRA*, 2018. 3

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015. 2

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE TPAMI*, 40(4): 834–848, 2018. 2

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 7, 8

[7] Po-Yi Chen, Alexander H. Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: unsupervised monocular depth estimation with semantic-aware representation. In *IEEE CVPR*, 2019. 3

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE CVPR*, 2016. 6, 7

[9] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *IEEE CVPR*, 2018. 2

[10] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *CACM*, 15 (1):11–15, 1972. 3

[11] Deng-Ping Fan, Ge-Peng Ji, Peng Xu, Ming-Ming Cheng, Christos Sakaridis, and Luc Van Gool. Advances in deep concealed scene understanding. *VI*, 1(1):16, 2023. 1

[12] Yasutaka Furukawa and Yoshihisa Shinagawa. Accurate and robust line segment extraction by analyzing distribution around peaks in hough space. *CVIU*, 92(1):1–25, 2003. 3

[13] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *IEEE CVPR*, 2019. 2

[14] Lukas Hoyer, Dengxin Dai, and Luc Gool. Daformer: improving network architectures and training strategies for domain-adaptive semantic segmentation. In *IEEE CVPR*, 2022. 6

[15] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *IEEE CVPR*, 2020. 2

[16] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. CCNet: Criss-cross attention for semantic segmentation. In *IEEE ICCV*, 2019. 2

[17] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *IEEE CVPR*, 2019. 2

[18] Zhenchao Jin, Tao Gong, Dongdong Yu, Qi Chu, Jian Wang, Changhu Wang, and Jie Shao. Mining contextual information beyond image for semantic segmentation. In *IEEE ICCV*, 2021. 2

[19] Zhenchao Jin, Bin Liu, Qi Chu, and Nenghai Yu. ISNet: Integrate image-level and semantic-level context for semantic segmentation. In *IEEE ICCV*, 2021. 2

[20] Jana Košecká and Wei Zhang. Efficient computation of vanishing points. In *IEEE ICRA*, 2002. 3

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25, 2012. 3

[22] Shih-Po Lee, Si-Cun Chen, and Wen-Hsiao Peng. GSVNet: Guided spatially-varying convolution for fast semantic segmentation on video. In *IEEE ICME*, 2021. 2

[23] Jiangtong Li, Wentao Wang, Junjie Chen, Li Niu, Jianlou Si, Chen Qian, and Liqing Zhang. Video semantic segmentation via sparse temporal transformer. In *ACM MM*, 2021. 1, 2

[24] Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *IEEE CVPR*, 2020. 2

[25] Xiangtai Li, Wenwei Zhang, Jiangmiao Pang, Kai Chen, Guangliang Cheng, Yunhai Tong, and Chen Change Loy. Video k-net: A simple, strong, and unified baseline for video segmentation. In *IEEE CVPR*, 2022. 7, 8

[26] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *IEEE CVPR*, 2018. 2

[27] Hyunjun Lim, Yeeun Kim, Kwangik Jung, Sumin Hu, and Hyun Myung. Avoiding degeneracy for monocular visual slam with point and line features. In *IEEE ICRA*, 2021. 3

[28] Jianbo Liu, Junjun He, Yu Qiao, Jimmy S Ren, and Hongsheng Li. Learning to predict context-adaptive convolution for semantic segmentation. In *ECCV*, 2020. 2

[29] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *ECCV*, 2020. 2, 3, 7, 8

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2017. 6

[31] Evelyne Lutton, Henri Maître, and Jaime Lopez Krahe. Contribution to the determination of vanishing points using hough transform. *IEEE TPAMI*, 16(4):430–438, 1994. 3

[32] Behrooz Mahasseni, Sinisa Todorovic, and Alan Fern. Budget-aware deep semantic video segmentation. In *IEEE CVPR*, 2017. 2

[33] Jiaxu Miao, Yunchao Wei, Yu Wu, Chen Liang, Guangrui Li, and Yi Yang. VSPW: A large-scale dataset for video scene parsing in the wild. In *IEEE CVPR*, 2021. 1, 2, 3, 7, 8

[34] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: a survey. *IEEE TPAMI*, 44 (7):3523–3542, 2022. 2

[35] Peyman Moghadam, Janusz A. Starzyk, and W. Sardha Wijesoma. Fast vanishing-point detection in unstructured environments. *IEEE TIP*, 21(1):425–430, 2012. 3

[36] Matthieu Paul, Martin Danelljan, Luc Van Gool, and Radu Timofte. Local memory attention for fast video semantic segmentation. In *IEEE IROS*, 2021. 2, 3

[37] 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 TPAMI*, 44(3), 2022. 8

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 6

[39] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *IEEE ICCV*, 2021. 6, 7

[40] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *ECCV*, 2016. 2

[41] Jinjin Shi, Jinxiang Wang, and Fangfa Fu. Fast and robust vanishing point detection for unstructured road following. *IEEE TITS*, 17(4):970–979, 2016. 3

[42] Guolei Sun, Yun Liu, Henghui Ding, Thomas Probst, and Luc Van Gool. Coarse-to-fine feature mining for video semantic segmentation. In *IEEE CVPR*, 2022. 1, 2, 3, 7, 8

[43] Guolei Sun, Yun Liu, Hao Tang, Ajad Chhatkuli, Le Zhang, and Luc Van Gool. Mining relations among cross-frame affinities for video semantic segmentation. In *ECCV*, 2022. 1, 2, 3, 7, 8

[44] Thorsten Suttorp and Thomas Bücher. Robust vanishing point estimation for driver assistance. In *IEEE ITSC*, 2006. 3

[45] Arces Talavera, Daniel Stanley Tan, Arnulfo Azcarraga, and Kai-Lung Hua. Layout and context understanding for image synthesis with scene graphs. In *IEEE ICIP*, 2019. 3

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 4, 5

[47] Qiang Wan, Zilong Huang, Jiachen Lu, Gang Yu, and Li Zhang. Seaformer: Squeeze-enhanced axial transformer for mobile semantic segmentation. In *ICLR*, 2023. 7, 8

[48] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation. In *IEEE ICIP*, 2021. 8

[49] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 2, 7

[50] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 3

[51] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 6, 7, 8

[52] Li Xingxin, Liqiang Zhu, Yu Zujun, and Wan Yanqin. Adaptive auxiliary input extraction based on vanishing point detection for distant object detection in high-resolution railway scene. In *IEEE ICEMI*, 2019. 3

[53] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. Dynamic video segmentation network. In *IEEE CVPR*, 2018. 2

[54] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. DenseASPP for semantic segmentation in street scenes. In *IEEE CVPR*, 2018. 2

[55] Ju Han Yoo, Seong-Whan Lee, Sung-Kee Park, and Dong Hwan Kim. A robust lane detection method based on vanishing point estimation using the relevance of line segments. *IEEE TITS*, 18(12):3254–3266, 2017. 3

[56] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2

[57] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 7

[58] Zhiliang Zeng, Mengyang Wu, Wei Zeng, and Chi-Wing Fu. Deep recognition of vnishing-point-constrained building planes in urban street views. *IEEE TIP*, 29:5912–5923, 2020. 3

[59] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *IEEE CVPR*, 2018. 2

[60] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE CVPR*, 2017. 7

[61] Mingmin Zhen, Jinglu Wang, Lei Zhou, Shiwei Li, Tianwei Shen, Jiaxiang Shang, Tian Fang, and Long Quan. Joint semantic segmentation and boundary detection using iterative pyramid contexts. In *IEEE CVPR*, 2020. 2

[62] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Context-reinforced semantic segmentation. In *IEEE CVPR*, 2019. 2

[63] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *IEEE CVPR*, 2017. 1, 2

[64] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *IEEE ICCV*, 2019. 2

[65] Zhaozi Zu, Yingtuan Hou, Dixiao Cui, and Jianru Xue. Real-time road detection with image texture analysis-based vanishing point estimation. In *IEEE PIC*, 2015. 3