Power of Boundary and Reflection: Semantic Transparent Object Segmentation using Pyramid Vision Transformer with Transparent Cues
— Supplementary Material —

Abstract

Our supplementary material has three sections. Section 1 shows the detailed architecture of each module in our proposed method and provides hyperparameters of each scale of PVTv1 and PVTv2. Section 2 contains the experimental setup, including datasets, implementation details, and evaluation metrics. Additional analysis is detailed in Section 3, together with quantitative and qualitative results of each dataset. In addition, we provide images and videos as a demo of deploying our method on a real robot.

1. Network architecture

In this section, we show the detailed architecture of the Feature Extraction Module in our encoder and the Feature Parsing Module in our decoder in Figure 1.

The hyperparameters of backbones in our models are listed as follows:

- S_i: stride of overlapping patch embedding in Stage i;
- C_i: channel number of output of Stage i;
- L_i: number of encoder layers in Stage i;
- R_i: reduction ratio of SRA layer in Stage i;
- P_i: patch size of Stage i;
- N_i: head number of Efficient Self-Attention in Stage i;
- E_i: expansion ratio of Feed-Forward layer [49] in Stage i;

In addition, we describe a series of PVTv1 [51] backbones with different scales (Tiny, Small, Medium, and Large) in Table 1 and a series of PVTv2 [52] backbones with different scales (B1 to B5) in Table 2.

2. Experimental Setups

2.1. Datasets

We comprehensively evaluated our proposed method on diverse datasets to demonstrate its exceptional performance and versatility. These datasets encompass a broad spectrum of segmentation tasks such as Glass (Transparent) datasets (Trans10k-v2 [60], RGBP-Glass [37], and GSD-S [28]), Mirror (Reflection) datasets (MSD [65], PMD [26], and RGBD-Mirror [36]), and generic datasets, which consists of both glass and mirror objects (TROSD [46], and Stanford2D3D [1]), ranging from binary to semantic segmentation, with a particular focus on images featuring reflective, transparent, or both characteristics. Our evaluation also considers the varied positions and fields of view (FOV) of objects within the images. Objects of interest may appear near or far from the camera's perspective, positioned randomly or at the center of the frame, providing a rich and realistic testing environment. Furthermore, the datasets we utilized are substantial in size, ensuring coverage of a broad range of environmental and scenario complexities. This encompasses indoor and outdoor scenarios, varying lighting conditions, diverse object scales, different viewpoints, and levels of occlusion. Our extensive evaluation showcases the robustness and adaptability of our method across a wide array of real-world conditions. Details of each dataset are shown in Table 3.

2.2. Implementation Details

We implemented our method in PyTorch 1.8.0 and CUDA 11.2. We adopted AdamW optimizer [33] where the learning rate γ was set to 10^{-4} with epsilon 10^{-8} and weight decay 10^{-4}. Our model was trained with a batch size of 8 and on a single NVIDIA RTX 3090 GPU, but it can still be trained on an older 2080 Ti or 1080 Ti GPU with a smaller batch size, e.g., 4. We evaluated all variants of our network on the validation set at every epoch during training. We used the best model of each variant on the validation set to evaluate the variant on the test set. The training process was completed once no further improvements were achieved. We use mean Intersection over Union (mIoU) as the primary evaluation metric to assess segmentation performance.

2.3. Evaluation metrics.

We adopt four widely used metrics from [37] to assess glass segmentation performance quantitatively: mean intersection over union (mIoU), weighted F-measure (F_β^w), mean absolute error (MAE), and balance error rate (BER).

Intersection over Union (IoU) is a widely used metric in

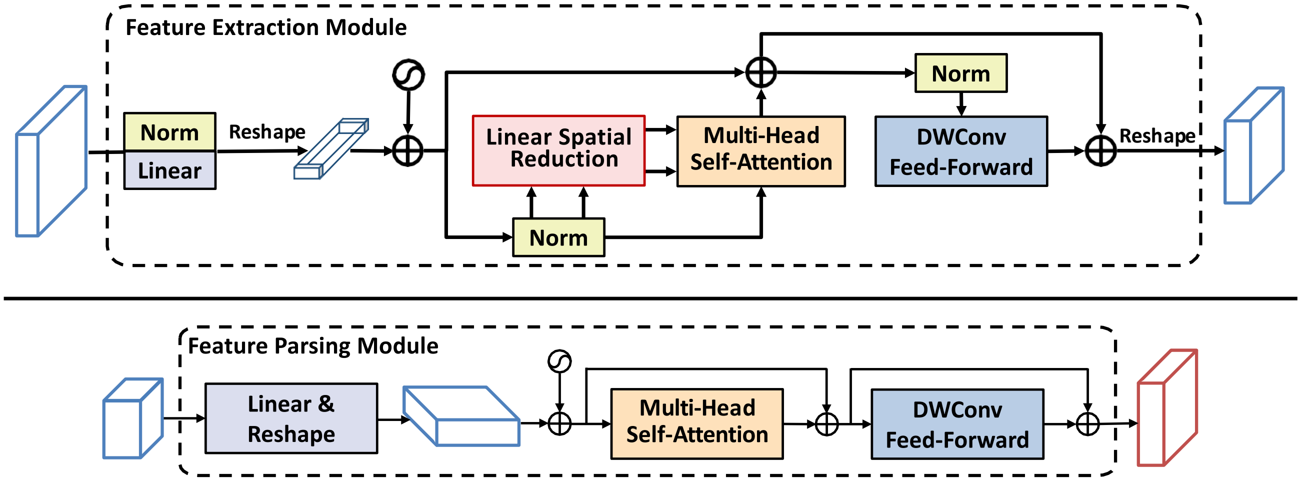

Figure 1. The architecture of the Feature Extraction Module (top) in our encoder and Feature Parsing Module (bottom) in our decoder. Zoom in for better visualization.

Table 1. Detailed settings of PVTv1 series which is adopted from [51].

	Output Size	**Layer Name**	**PVT-Tiny**	**PVT-Small**	**PVT-Medium**	**PVT-Large**
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	Patch Embedding	$P_1 = 4;\ C_1 = 64$			
		Transformer Encoder	$\begin{bmatrix} R_1 = 8 \\ N_1 = 1 \\ E_1 = 8 \end{bmatrix} \times 2$	$\begin{bmatrix} R_1 = 8 \\ N_1 = 1 \\ E_1 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} R_1 = 8 \\ N_1 = 1 \\ E_1 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} R_1 = 8 \\ N_1 = 1 \\ E_1 = 8 \end{bmatrix} \times 3$
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	Patch Embedding	$P_2 = 2;\ C_2 = 128$			
		Transformer Encoder	$\begin{bmatrix} R_2 = 4 \\ N_2 = 2 \\ E_2 = 8 \end{bmatrix} \times 2$	$\begin{bmatrix} R_2 = 4 \\ N_2 = 2 \\ E_2 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} R_2 = 4 \\ N_2 = 2 \\ E_2 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} R_2 = 4 \\ N_2 = 2 \\ E_2 = 8 \end{bmatrix} \times 8$
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	Patch Embedding	$P_3 = 2;\ C_3 = 320$			
		Transformer Encoder	$\begin{bmatrix} R_3 = 2 \\ N_3 = 5 \\ E_3 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} R_3 = 2 \\ N_3 = 5 \\ E_3 = 4 \end{bmatrix} \times 6$	$\begin{bmatrix} R_3 = 2 \\ N_3 = 5 \\ E_3 = 4 \end{bmatrix} \times 18$	$\begin{bmatrix} R_3 = 2 \\ N_3 = 5 \\ E_3 = 4 \end{bmatrix} \times 27$
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	Patch Embedding	$P_4 = 2;\ C_4 = 512$			
		Transformer Encoder	$\begin{bmatrix} R_4 = 1 \\ N_4 = 8 \\ E_4 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} R_4 = 1 \\ N_4 = 8 \\ E_4 = 4 \end{bmatrix} \times 3$	$\begin{bmatrix} R_4 = 1 \\ N_4 = 8 \\ E_4 = 4 \end{bmatrix} \times 3$	$\begin{bmatrix} R_4 = 1 \\ N_4 = 8 \\ E_4 = 4 \end{bmatrix} \times 3$

segmentation tasks, which is defined as:

$$IoU = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(G(i,j) * P_b(i,j))}{\sum_{i=1}^{H}\sum_{j=1}^{W}(G(i,j) + P_b(i,j) - G(i,j) * P_b(i,j))} \tag{1}$$

where G is the ground truth mask with values of the glass region being one while those of the non-glass region are 0; P_b is the predicted mask binarized with a threshold of 0.5; and H and W are the height and width of the ground truth mask, respectively.

Weighted F-measure (F_β^w) is adopted from the salient object detection tasks with $\beta = 0.3$. F-measure (F_β) is a measure of both the precision and recall of the prediction map. Recent studies [9] have suggested that the weighted F-measure (F_β^w) [34] can provide more reliable evaluation results than the traditional F_β. Thus, we report F_β^w in the comparison.

Mean Absolute Error (MAE) is widely used in foreground-background segmentation tasks, which calculates the element-wise difference between the prediction map P and the ground truth mask G :

$$MAE = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}|P(i,j) - G(i,j)|, \tag{2}$$

Table 2. Detailed settings of PVTv2 series which is adopted from [52].

	Output Size	Layer Name	PVT-B1	PVT-B2	PVT-B3	PVT-B4	PVT-B5
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	Overlapping Patch Embedding	$S_1 = 4$				
			$C_1 = 64$				
		Transformer Encoder	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 2$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 3$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 3$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 3$	$R_1 = 8$ $N_1 = 1$ $E_1 = 4$ $L_1 = 3$
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	Overlapping Patch Embedding	$S_2 = 2$				
			$C_2 = 128$				
		Transformer Encoder	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 2$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 3$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 3$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 8$	$R_2 = 4$ $N_2 = 2$ $E_2 = 4$ $L_2 = 6$
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	Overlapping Patch Embedding	$S_3 = 2$				
			$C_3 = 320$				
		Transformer Encoder	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 2$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 6$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 18$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 27$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 40$
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	Overlapping Patch Embedding	$S_4 = 2$				
			$C_4 = 512$				
		Transformer Encoder	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 2$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$

Table 3. Comparison between different datasets in our experiments. “P”, “D”, and “S” denote polarization images, depth images, and semantic maps, respectively. Note that our method **uses only RGB images** as input for both training and testing.

	Dataset	Modalities	No. of Images	Tasks	Types	FOV	Position
Glass	Trans10k-v2 [60]	RGB	10,428	semantic	both	both	random
	RGBP-Glass [37]	RGB-P	4,511	binary	transparent	far	random
	GSD-S [28]	RGB-S	4,519	binary	transparent	far	random
Mirror	MSD [65]	RGB	4,018	binary	reflective	both	center
	PMD [26]	RGB	6,461	binary	reflective	both	random
	RGBD-Mirror [36]	RGB-D	3,049	binary	reflective	both	center
Generic	TROSD [46]	RGB-D	11,060	semantic	both	near	center
	Stanford2D3D [1]	RGB-D	70,496	semantic	both	far	random

where $P(i, j)$ indicates the predicted probability score at location (i, j).

Balance Error Rate (BER) is a standard metric used in shadow detection tasks, defined as:

$$BER = (1 - \frac{1}{2}(\frac{TP}{N_p} + \frac{TN}{N_n})) \times 100 \qquad (3)$$

where TP, TN, N_p, and N_n represent the numbers of true positive, true negative, glass, and non-glass pixels, respectively.

Table 4. Quantitative comparison against SOTAs on RGB-P dataset [37]. (∗) denotes the glass segmentation methods with additional input polarization images.

Method	GFLOPs ↓	mIoU ↑	F_β^w ↑	MAE ↓	BER ↓
EAFNet [58] ∗	18.93	53.86	0.611	0.237	24.65
PM R-CNN [21] ∗	56.59	66.03	0.714	0.178	18.92
PGSNet [37] ∗	290.62	81.08	0.842	0.091	9.63
Trans2Seg [60]	49.03	75.21	0.799	0.122	13.23
TransLab [59]	61.26	73.59	0.772	0.148	15.73
SegFormer [61]	70.24	78.42	0.815	0.121	13.03
GSD [27]	92.69	78.11	0.806	0.122	12.61
Ours-B5	154.37	82.77	0.879	**0.042**	**9.59**
GDNet [35]	271.53	77.64	0.807	0.119	11.79
SETR [75]	240.11	77.60	0.817	0.114	11.46
PanoGlassNet [2]	**581.04**	**86.89**	**0.929**	0.068	-

2.4. Qualitative and Quantitative Results

We evaluated the performance of our method across three distinct tasks: glass segmentation, mirror segmentation, and generic segmentation. To ensure **fair comparisons**, we have carefully selected our model variants (Ours-X with X is postfixes: -T, -S, -M, -L, -B1, -B2, -B3, -B4, and -B5, represented the size of the model as PVTv1 Tiny, Small, Medium, Large, and PVTv2 B1-5, respectively) that have **similar model's size or complexity** used by other methods, as indicated in the respective tables.

3. Additional Experiments

3.1. Comparison on Glass Object Segmentation

We benchmarked our method against recent glass segmentation methods on the binary (RGBP-Glass and GSD-S dataset) and semantic segmentation (Trans10K-v2 dataset) tasks.

RGBP-Glass dataset. We extensively compare the effectiveness of our method with state-of-the-art methods, as shown in Table 4. All methods are retrained on the RGBP-Glass dataset [37] for a fair comparison. EAFNet [58], Polarized Mask R-CNN (P.M. R-CNN) [21], and PGSNet [37] are the three methods that leverage polarization cues. SETR [75], SegFormer [61] are the two methods focusing on general semantic/instance segmentation tasks. GDNet [35], TransLab [59], Trans2Seg [60], and GSD [27] and our method are in-the-wild glass segmentation methods but only rely on RGB input. From Figure 2, we can see that our method outperforms all other methods. It should be noted that our method outperforms previous works that utilize additional input signals, such as polarization cues [21, 37, 58], while remaining efficient.

Table 5. Evaluation results on GSD-S dataset [28]. **Note that:** we use only RGB as input to our method. (†) denotes the glass segmentation method with additional semantic context information and post-processing refinement.

Method	mIoU ↑	F_β^w ↑	MAE ↓	BER ↓
PSPNet [72]	56.1	0.679	0.093	13.41
DeepLabV3+ [5]	55.7	0.671	0.100	13.11
PSANet [74]	55.1	0.656	0.104	12.61
DANet [11]	54.3	0.673	0.098	14.78
SCA-SOD [43]	55.8	0.689	0.087	15.03
SETR [75]	56.7	0.679	0.086	13.25
Segmenter [45]	53.6	0.645	0.101	14.02
Swin [31]	59.6	0.702	0.082	11.34
T-2-T ViT [67]	56.2	0.693	0.087	14.72
SegFormer [61]	54.7	0.683	0.094	15.15
Twins [6]	59.1	0.703	0.084	12.43
GDNet [35]	52.9	0.642	0.101	18.17
GSD [27]	72.1	0.821	0.061	10.02
VBNet [40]	73.5	0.837	0.038	10.07
Ours-B5	75.2	0.859	0.046	**9.04**
GlassSemNet [28] †	**75.3**	**0.860**	**0.035**	9.26

GSD-S dataset. We compare our method with other recent methods in Table 5 and Figure 3, includes generic semantic segmentation methods (PSPNet [72], DeepLabV3+ [5], PSANet [74], DANet [11]), recent state-of-the-art models that utilize transformer technique (SETR [75], Swin [31], SegFormer [61], Twins [6]), and glass surface detection methods (GDNet [35], GSD [27], GlassSemNet [28]). For a fair comparison, all methods are retrained on the GSD-S dataset [28]. Our method outperforms all other methods and achieves comparable performance to GlassSemNet [28], which provides additional semantic context information. GlassSemNet [28] points out that humans frequently use the semantic context of their surroundings to reason, as this provides information about the types of things to be found and how close they might be to one another. For instance, glass windows are more likely to be found close to other semantically related objects (walls and curtains) than to things (cars and trees). Their method uses semantic context information as an additional input to progressively learn contextual correlations among objects, both spatially and semantically, thereby boosting performance. Their predictions are then refined by Fully Connected Conditional Random Fields [22] to improve performance further.

Trans10k-v2 dataset. Shifting our focus to the semantic glass segmentation task, where the challenge extends beyond merely detecting glass areas to classifying them into 11 fine-grained categories, our method still reigns supreme, as shown in Table 6. Figure 4 also confirms that our method achieves higher segmentation quality with better transparent features, e.g., the segmentation of two overlapping doors is accurately obtained. These comprehensive evaluations un-

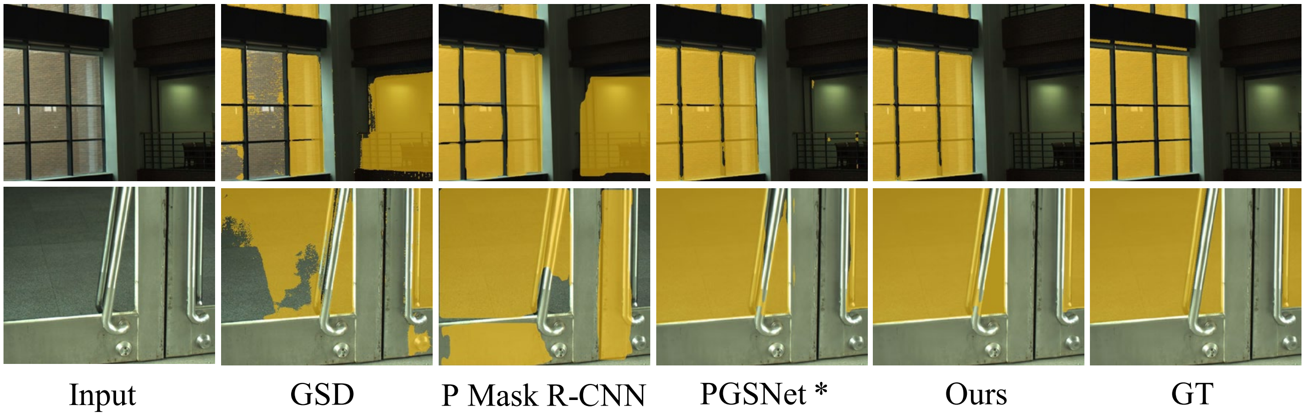

Figure 2. Qualitative comparison of our method with other methods on RGB-P dataset [37]. (∗) denotes the glass segmentation method with additional polarization images as input.

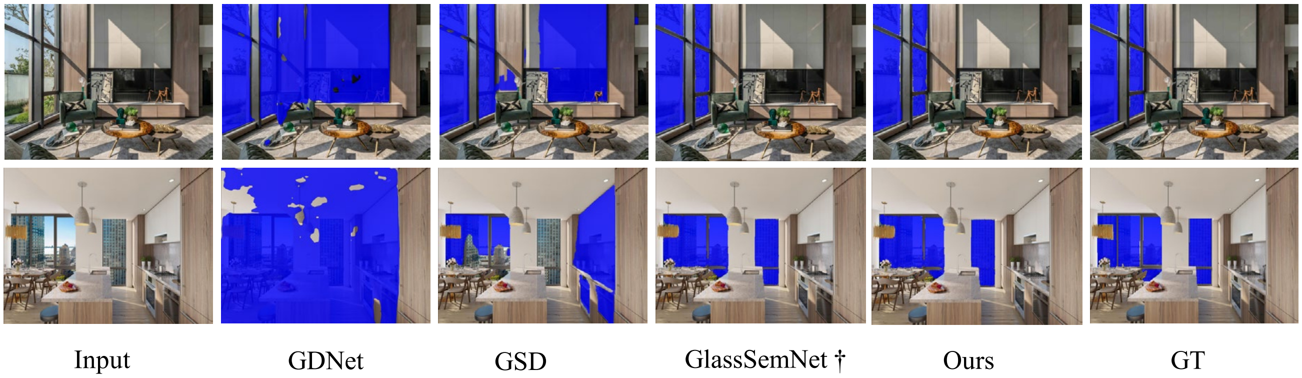

Figure 3. Qualitative comparison of our method with other methods on GSD-S dataset [28]. (†) denotes the glass segmentation method with semantic context and post-processing refinement.

derscore the effectiveness of our approach across diverse glass segmentation scenarios, affirming its position as a top-performing and computationally efficient choice for these tasks.

3.1.1. Comparison on Binary Mirror Segmentation

MSD and PMD datasets. We compare quantitative results of the state-of-the-art methods and our method on MSD and PMD datasets, including four RGB salient object detection methods CPDNet [57], MINet [39], LDF [56], and VST [30], and five mirror detection methods MirrorNet [65], PMDNet [26], SANet [13], VCNet [47], SATNet [18]. As shown in Table 7, our method achieves the best performance in terms of all the evaluation metrics. Significantly, we outperform the second-best method by 5.63% on the MSD dataset.

RGBD-Mirror dataset. Our method is also compared with seven RGB-D salient object detection methods such as HDFNet [38], S2MA [29], JL-DCF [12], DANet [11], BBSNet [10] and VST [30], and four mirror detection methods, including PDNet [36], SANet [13], VCNet [47], and PDNet [36] on the RGBD-Mirror dataset. Our method outperforms all competing methods, even though we do not use depth information, as shown in Table 8.

3.1.2. Comparison on Generic Segmentation

Stanford2D3D dataset. As shown in Table 9, we compare with other methods across different backbone sizes. Our method outperforms existing work by about 10.1% in mIOU, highlighting the segmentation capacity of our network in general scenes with glass objects.

TROSD dataset. We compared our method with SOTAs on the TROSD dataset [46], a dataset specifically for transparent and reflective objects. Table 10 provides an overview of our competitors and highlights their best results, achieved using their publicly available source codes. All methods utilized the same data augmentation strategy.

ADE20k and Cityscapes datasets. We conducted ad-

Table 6. Quantitative evaluation of our method and existing methods on the Trans10K-v2 dataset [60].

Method	GFLOPs ↓	MParams ↓	ACC ↑	mIoU ↑
HRNet_w18 [50]	4.20	1.53	89.58	54.25
LEDNet [54]	6.23	-	86.07	46.40
Trans4Trans-T [71]	10.45	-	<u>93.23</u>	<u>68.63</u>
Ours-T	10.50	12.72	**93.52**	**69.53**
ICNet [73]	10.64	8.46	78.23	23.39
BiSeNet [66]	19.91	13.3	89.13	58.40
Trans4Trans-S [71]	19.92	-	94.57	74.15
Ours-S	20.00	23.98	<u>94.83</u>	<u>75.32</u>
Ours-B1	21.29	14.87	**95.37**	**77.05**
Trans4Trans-M [71]	34.38	-	95.01	75.14
Ours-M	34.51	43.70	95.08	76.06
DenseASPP [64]	36.20	29.09	90.86	63.01
Ours-B2	37.03	27.59	**95.92**	**79.29**
DeepLabv3+ [5]	37.98	28.74	92.75	68.87
FCN [32]	42.23	34.99	91.65	62.75
RefineNet [25]	44.56	29.36	87.99	58.18
Trans2Seg [60]	49.03	56.20	94.14	72.15
Ours-L	50.54	60.86	<u>95.28</u>	<u>77.35</u>
TransLab [59]	61.31	42.19	92.67	69.00
Ours-B3	68.35	51.21	<u>96.28</u>	<u>80.04</u>
Ours-B4	79.34	67.11	**96.59**	**80.99**
To-Former-B2 [4]	117.74	-	-	77.43
U-Net [41]	124.55	13.39	81.90	29.23
DUNet [20]	123.69	-	90.67	59.01
Ours-B5	154.37	106.19	**96.93**	**81.37**
DANet [11]	198.00	-	<u>92.70</u>	<u>68.81</u>
PSPNet [72]	187.03	50.99	92.47	68.23

ditional experiments on the ADE20K and CityScapes datasets, with the results (mIoU) shown in Table 11, sorted in ascending order of GFLOPs (512×512). As can be seen, our method performs well on both datasets, with mIoU 47.5% on ADE20K and 81.9% on CityScapes.

3.2. Ablation studies

We present additional ablation studies to verify various aspects of our model's design.

Different combinations of network architecture. Table 12 presents comparisons among various combinations of encoders and decoders, such as using only a CNN architecture, using a combination of CNN and Transformer, and using a fully Transformer-based model. Our method, an encoder-decoder transformer-based model, outperforms competitive networks, indicating the system's capability to segment transparent objects effectively. In this ablation study, we used Ours-M and Ours-B2 (not the best model, Ours-B5), which have the same network size as other methods (-M model size), for a fair comparison.

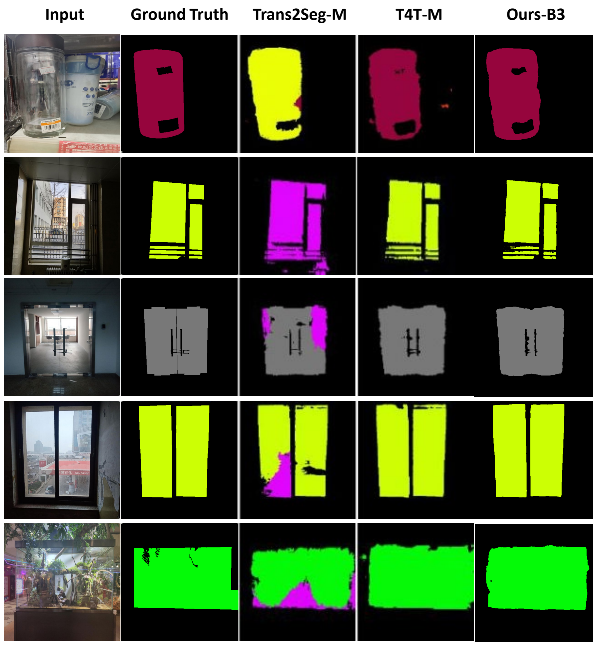

Figure 4. Qualitative comparison of our method and existing methods on Trans10K-v2 [60]. For a fair comparison, we used Ours-B3, which has the same network size as other methods (-M model).

Table 7. Quantitative results of our method with SOTAs on Salient Object Detection (the first five methods) and Mirror Detection (the last ten methods) on MSD and PMD datasets.

Method	MSD			PMD		
	IoU ↑	F_β^w ↑	MAE ↓	IoU ↑	F_β^w ↑	MAE ↓
CPDNet [57]	57.58	0.743	0.115	60.04	0.733	0.041
MINet [39]	66.39	0.823	0.087	60.83	0.798	0.037
LDF [56]	72.88	0.843	0.068	63.31	0.796	0.037
VST [30]	79.09	0.867	0.052	59.06	0.769	0.035
ShadowSAM [55]	-	0.700	0.080	-	0.685	0.095
MirrorNet [65]	78.88	0.856	0.066	58.51	0.741	0.043
PMDNet [26]	81.54	0.892	0.047	66.05	0.792	0.032
SANet [13]	79.85	0.879	0.054	66.84	0.837	0.032
HetNet [17]	82.80	0.906	0.043	69.00	0.814	0.029
UTLNeT [77]	83.05	0.892	0.040	-	-	-
WSMD [70]	75.00	0.780	0.078	60.00	0.630	0.051
DPRNet [69]	<u>86.60</u>	0.888	0.033	**72.10**	0.766	0.026
VCNet [47]	80.08	0.898	0.044	64.02	0.815	0.028
SATNet [18]	85.41	<u>0.922</u>	0.033	<u>69.38</u>	<u>0.847</u>	<u>0.025</u>
CSFwinformer [62]	-	0.865	<u>0.030</u>	-	0.836	0.039
Ours-B3	**91.04**	**0.953**	**0.028**	<u>69.61</u>	**0.853**	**0.021**

Analysis of different backbones. We have conducted experiments using alternative backbones, as presented in Figure 5. Among these options, the PVT-v2 backbone [52] stands out with significantly higher mIoU and remarkably compact model size (MParams). Despite its higher GFLOP complexity compared to the FocalNet backbone [63], it still achieves better performance. Additionally, the PVT-

Table 8. Quantitative results of SOTAs on RGBD-Mirror dataset.

Method	Input	IoU ↑	F^w_β ↑	MAE ↓
HDFNet [38]	RGB-D	44.73	0.733	0.093
S2MA [29]	RGB-D	60.87	0.781	0.070
DANet [11]	RGB-D	67.81	0.835	0.060
JL-DCF [12]	RGB-D	69.65	0.844	0.056
VST [30]	RGB-D	70.20	0.851	0.052
BBSNet [10]	RGB-D	74.33	0.868	0.046
PDNet [36]	RGB-D	77.77	0.878	0.041
UTLNet [77]	RGB-D	80.50	0.858	0.032
VCNet [47]	RGB	73.01	0.849	0.052
PDNet [36]	RGB	73.57	0.851	0.053
SANet [13]	RGB	74.99	0.873	0.048
SATNet [18]	RGB	78.42	0.906	0.031
WSMD [70]	RGB	61.60	0.655	0.088
DPRNet [69]	RGB	76.10	0.811	0.047
Ours-B3	RGB	**88.52**	**0.954**	**0.027**

Table 9. Comparison with SOTAs on Stanford2D3D dataset.

Method	GFLOPs ↓	MParams ↓	mIoU ↑
PVT-T [51]	10.16	13.11	41.00
Trans4Trans-T [71]	10.45	12.71	41.28
Ours-T	10.50	12.72	47.11
Trans2Seg-T [60]	16.96	17.87	42.07
Ours-B1	21.99	14.87	**51.55**
PVT-S [51]	19.58	24.36	41.89
Trans4Trans-S [71]	19.92	23.95	44.47
Ours-S	20.00	23.98	50.17
Trans2Seg-S [60]	30.26	27.98	42.91
Ours-B2	37.03	27.59	**53.98**
Trans4Trans-M [71]	34.38	43.65	45.73
Ours-M	34.51	43.70	52.57
Trans2Seg-M [60]	40.98	30.53	43.83
PVT-M [51]	49.00	56.20	42.49
Ours-B3	68.35	51.21	**54.66**
Ours-L	50.54	60.86	53.75
Ours-B4	79.34	67.11	**55.21**
Ours-B5	154.37	106.19	**55.83**

Table 10. Comparison of different methods on TROSD. R: reflective objects. T: transparent objects. B: background.

Method	Input	IOU ↑			mIoU ↑	mAcc ↑
		R	T	B		
RefineNet [25]	RGB	21.32	37.32	92.37	50.34	63.59
ANNNet [78]	RGB	22.31	41.30	93.43	52.35	62.49
Trans4Trans [71]	RGB	27.69	39.22	94.16	53.69	61.82
PSPNet [72]	RGB	26.35	44.38	94.19	54.97	64.14
OCNet [68]	RGB	31.76	46.52	95.05	57.78	64.46
TransLab [59]	RGB	42.57	50.72	96.01	63.11	68.72
DANet [11]	RGB	42.76	54.39	95.88	64.34	70.95
TROSNet [46]	RGB	48.75	48.56	95.49	64.26	75.93
Ours	RGB	**66.16**	**66.83**	**97.71**	**76.90**	**87.62**
SSMA [48]	RGB-D	24.70	29.04	89.98	47.91	67.72
FRNet [76]	RGB-D	28.37	36.59	92.18	52.38	63.94
EMSANet [42]	RGB-D	27.53	44.10	96.14	55.92	71.63
FuseNet [15]	RGB-D	37.30	43.29	94.97	58.52	66.13
RedNet [19]	RGB-D	48.27	47.57	95.76	63.87	69.23
EBLNet [16]	RGB-D	51.75	50.12	94.57	65.49	67.39
TROSNet [46]	RGB-D	62.27	57.23	96.52	72.01	81.21

Table 11. Comparison (mIoU ↑) with SOTAs on ADE20k and Cityscapes (CitySc.) datasets.

Method	GFLOPs↓	MParams↓	Backbone	ADE20K	CitySc.
Trans4Trans-M [52]	41.9	49.6	PVTv2-B3	-	69.3
Semantic FPN [71]	62.4	49.0	PVTv2-B3	47.3	-
Ours-B3	68.3	51.2	PVTv2-B3	47.5	81.9
MogaNet-S [24]	189	29.0	SemFPN	47.7	-
NAT-Mini [14]	900	50.0	UPerNet	46.4	-
InternImage-T [53]	944	59.0	UPerNet	**47.9**	**82.5**

Table 12. Effectiveness of different network architecture combinations. Models are evaluated on the Trans10K-v2 dataset [60]. **Note that:** the results are sorted by ascending of GFLOPS.

Method	Encoder		Decoder		GFLOPs	mIoU
	Trans.	CNN	Trans.	CNN		
Trans4Trans-M [71]	✓		✓		34.3	75.1
Ours-M	✓		✓		34.5	76.1
Ours-B2	✓		✓		37.0	**79.3**
Trans2Seg-M [60]		✓	✓		40.9	69.2
FCN [32]		✓		✓	42.2	62.7
OCNet [68]		✓		✓	43.3	66.3
PVT-M [51]	✓		✓		49.0	72.1

v2 backbone [52] demonstrates a lower complexity than the DaViT backbone [7] while maintaining competitive mIoU results. These findings highlight the superiority of the PVT-v2 backbone [52] in achieving an optimal balance between performance and model size, making it a promising choice for our method. When comparing PVT-v1 [51] with other backbones, the PVT-v1 [51] backbone boasts a considerably smaller model size and lower complexity. Despite these advantages, its performance remains competitive with other backbones. This demonstrates the efficiency of the PVT-v1 backbone [51], which delivers comparable performance while being lighter and less computationally demanding.

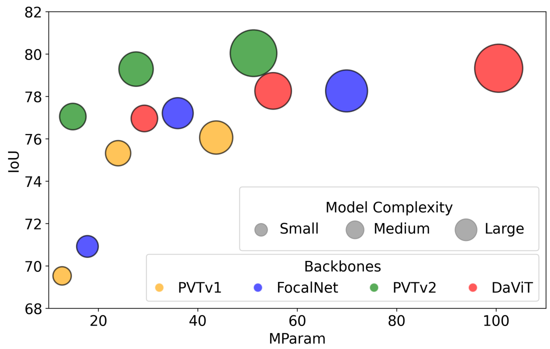

Figure 5. Our method with various backbones on Trans10K-v2 dataset. The bubble's size is its complexity in GFLOPs.

3.3. Further Analysis and Discussions

Effectiveness of the embedding channel. We experiment with the embedding channel with various values (64, 128, 256, 512) and report the mIoU and Accuracy of Ours-B1 model in Figure 6. Throughout the results, we show that our model achieved better performance with a higher number of embedding channels (from 77.05% at 64 channels to 78.85% at 512 channels). Note that, due to memory limits, we cannot perform experiments with higher embedding channels, *e.g.*, 1024 or 2048, and to save computational resources, we used Ours-B1 in this ablation study.

Real-time performance. We report the inference speed of our models on different GPUs (NVIDIA GTX 1070, NVIDIA RTX 3090) at a resolution of 512 × 512 and a batch size of 1. As shown in Figure 7, while Our-T model has a lower computational cost than other versions, it's important to note that all these models deliver performance levels well-suited for deployment on robotic systems. In real-world situations, achieving a similar level of prediction accuracy for each frame is crucial because it enables a navigation system to be more responsive, improving its capacity to assist robots efficiently.

Incorporation with other modalities. Integrating our model with depth images (RGB-D) or polarization images (RGB-P) is a feasible enhancement. A naive method involves adding an extra encoder to extract features from depth or polarization data. These additional features would then be fused with RGB features before our FPM module. This strategy is in line with PDNet [36], TROSNet [46], and PGSNet [37], as detailed in the supplementary material. Notably, including depth or polarization data in these models has led to significant performance gains and increased computational costs. Specifically, with added depth information, PDNet and TROSNet improved by +4% and +8% mIoU, and with added polarization information, PGSNet experienced a +5% boost in mIoU.

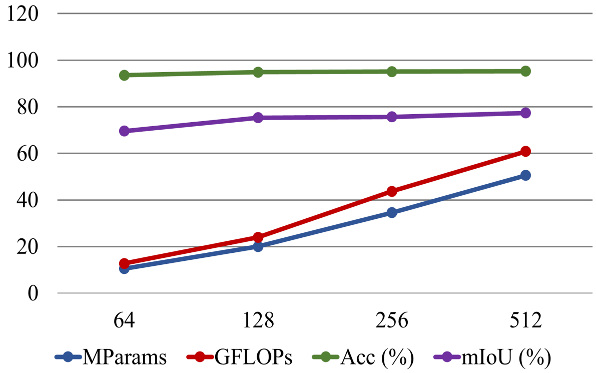

Figure 6. Effectiveness of the embedding channel in our method on Trans10K-v2 [60].

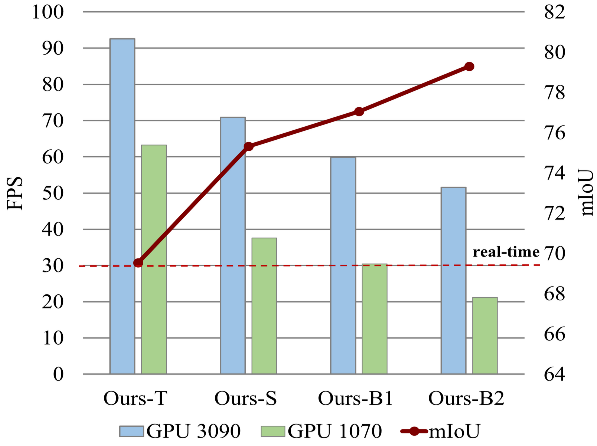

Figure 7. Inference time (FPS) of our method on Trans10K-v2.

Utilizing reflection removal methods for detecting reflections. Employing reflection removal techniques, as discussed in recent studies [8, 44], offers the potential to generate pseudo labels with distinct advantages. However, these methods are mainly designed to address global reflections when an image is entirely encompassed by glass. These methods have limitations in complex real-world situations in which glass objects are distributed throughout the scene rather than occupying a dominant position. Our study introduces the RFE module, which can detect localized reflections and distinguish glass surfaces based on the semantic mask. This module is better suited to the diverse and unpredictable conditions found in real-world situations, where reflections are specific to certain areas rather than uniformly distributed across the entire image, making it a better fit for real-world scenarios.

Comparison with foundation models. To fully evaluate our method's performance, we also compared it with recent powerful foundation models, such as the SAM model, and the results are shown in Figure 8. It is important to note that the SAM model **does not include semantics, or in other**

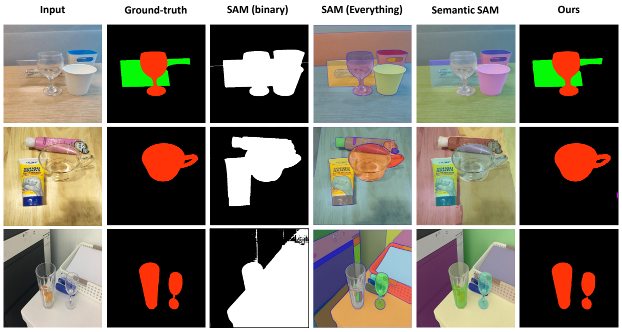

Figure 8. Qualitative comparison of our method with recent foundation models on Trans10k-v2 dataset.

words, it cannot yield masks with semantics. The SAM model also presents the challenge of over-segmentation, thereby increasing the likelihood of false positives. As a result, we see that the SAM model (binary and everything) cannot distinguish between glass and non-glass regions, unlike our method. It is the same for SAM variants with semantics [3, 23], which still fail and cannot generate reasonable semantics either.

Further analysis on our reflection RFE module. To provide further verification of the effectiveness of the RFE module, we conducted the following additional experiments:

- We take a model with the RFE module that has already been trained. To demonstrate the effectiveness of RFE, we compare the feature maps before and after passing through the RFE module. In Figure 9, we can see that after passing through the RFE module, we can get a stronger reflection signal, such as the transparent glass area or the specular reflection appearing at the base of the wine glass.
- Using the same model, we try turning off the RFE module at inference by passing the feature map before RFE directly to the next step. Note that at training, the RFE module is *well-trained as usual*. Figure 10 shows that bypassing RFE results in a noisy feature map and wrong mask prediction. This means that our learning of RFE does not yield a trivial function, *e.g.* the identity, and that RFE plays an important role in processing the feature maps and output reflection masks.

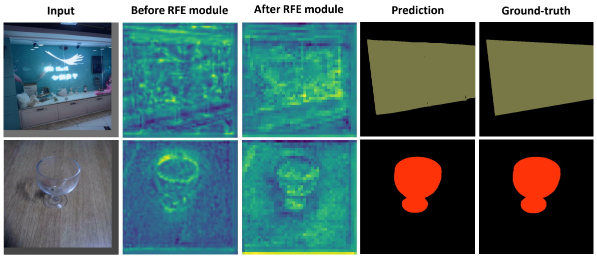

Figure 9. Comparison of the feature maps before and after passing through the RFE module on the Trans10k-v2 dataset.

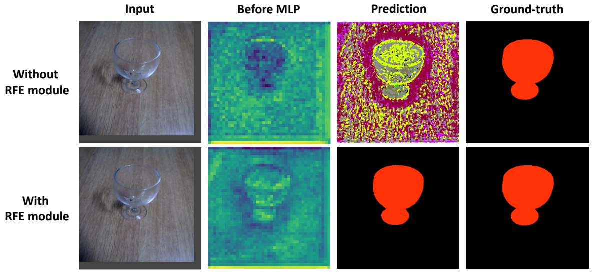

Figure 10. Comparison of the feature maps at inference by passing through the RFE module as usual (top) and bypassing the RFE module (bottom) on the Trans10k-v2 dataset.

References

[1] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *ArXiv e-prints*, 2017. 1, 3

[2] Qingling Chang, Huanhao Liao, Xiaofei Meng, Shiting Xu, and Yan Cui. Panoglassnet: Glass detection with panoramic rgb and intensity images. *IEEE Transactions on Instrumentation and Measurement*, 73:1–15, 2024. 4

[3] Jiaqi Chen, Zeyu Yang, and Li Zhang. Semantic segment anything. `https://github.com/fudan-zvg/Semantic-Segment-Anything`, 2023. 9

[4] Jiawei Chen, Wen Su, Mengjiao Ge, Ye He, and Jun Yu. To-former: semantic segmentation of transparent object with edge-enhanced transformer. *Vis. Comput.*, 41(3):1811–1825, 2024. 6

[5] 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. 4, 6

[6] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS*, 2021. 4

[7] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *ECCV*, 2022. 7

[8] Zheng Dong, Ke Xu, Yin Yang, Hujun Bao, Weiwei Xu, and Rynson W.H. Lau. Location-aware single image reflection removal. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4997–5006, 2021. 8

[9] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, 2017. 2

[10] Deng-Ping Fan, Yingjie Zhai, Ali Borji, Jufeng Yang, and Ling Shao. Bbs-net: Rgb-d salient object detection with a bifurcated backbone strategy network. In *ECCV*, 2020. 5, 7

[11] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019. 4, 5, 6, 7

[12] Keren Fu, Deng-Ping Fan, Ge-Peng Ji, and Qijun Zhao. Jl-dcf: Joint learning and densely-cooperative fusion framework for rgb-d salient object detection. In *CVPR*, 2020. 5, 7

[13] Huankang Guan, Jiaying Lin, and Rynson W.H. Lau. Learning semantic associations for mirror detection. In *CVPR*, 2022. 5, 6, 7

[14] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *CVPR*, 2023. 7

[15] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: incorporating depth into semantic segmentation via fusion-based cnn architecture. In *ACCV*, 2016. 7

[16] Hao He, Xiangtai Li, Guangliang Cheng, Jianping Shi, Yunhai Tong, Gaofeng Meng, Véronique Prinet, and LuBin Weng. Enhanced boundary learning for glass-like object segmentation. In *ICCV*, 2021. 7

[17] Ruozhen He, Jiaying Lin, and R.W.H. Lau. Efficient mirror detection via multi-level heterogeneous learning. In *AAAI*, 2023. 6

[18] Tianyu Huang, Bowen Dong, Jiaying Lin, Xiaohui Liu, Rynson W.H. Lau, and Wangmeng Zuo. Symmetry-aware transformer-based mirror detection. In *AAAI*, 2023. 5, 6, 7

[19] Jindong Jiang, Lunan Zheng, Fei Luo, and Zhijun Zhang. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. *arXiv preprint*, 2018. 7

[20] Qiangguo Jin, Zhaopeng Meng, Tuan D. Pham, Qi Chen, Leyi Wei, and Ran Su. DUNet: A deformable network for retinal vessel segmentation. *Knowledge-Based Systems*, 2019. 6

[21] Agastya Kalra, Vage Taamazyan, Supreeth Krishna Rao, Kartik Venkataraman, Ramesh Raskar, and Achuta Kadambi. Deep polarization cues for transparent object segmentation. In *2020 CVPR*, 2020. 4

[22] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. 4

[23] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity, 2023. 9

[24] Siyuan Li, Zedong Wang, Zicheng Liu, Cheng Tan, Haitao Lin, Di Wu, Zhiyuan Chen, Jiangbin Zheng, and Stan Z. Li. Efficient multi-order gated aggregation network. *ArXiv*, abs/2211.03295v2, 2023. 7

[25] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 6, 7

[26] Jiaying Lin, Guodong Wang, and Rynson W.H. Lau. Progressive mirror detection. In *CVPR*, 2020. 1, 3, 5, 6

[27] Jiaying Lin, Zebang He, and Rynson W.H. Lau. Rich context aggregation with reflection prior for glass surface detection. In *CVPR*, 2021. 4

[28] Jiaying Lin, Yuen-Hei Yeung, and R.W.H. Lau. Exploiting semantic relations for glass surface detection. *NeurIPS*, 2022. 1, 3, 4, 5

[29] Nian Liu, Ni Zhang, and Junwei Han. Learning selective self-mutual attention for rgb-d saliency detection. In *CVPR*, 2020. 5, 7

[30] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *ICCV*, 2021. 5, 6, 7

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 4

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 6, 7

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 1

[34] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *CVPR*, 2014. 2

[35] Haiyang Mei, Xin Yang, Yang Wang, Yuanyuan Liu, Shengfeng He, Qiang Zhang, Xiaopeng Wei, and Rynson W.H. Lau. Don't hit me! glass detection in real-world scenes. In *CVPR*, 2020. 4

[36] Haiyang Mei, Bo Dong, Wen Dong, Pieter Peers, Xin Yang, Qiang Zhang, and Xiaopeng Wei. Depth-aware mirror segmentation. In *CVPR*, 2021. 1, 3, 5, 7, 8

[37] Haiyang Mei, Bo Dong, Wen Dong, Jiaxi Yang, Seung-Hwan Baek, Felix Heide, Pieter Peers, Xiaopeng Wei, and Xin Yang. Glass segmentation using intensity and spectral polarization cues. In *CVPR*, 2022. 1, 3, 4, 5, 8

[38] Youwei Pang, Lihe Zhang, Xiaoqi Zhao, and Huchuan Lu. Hierarchical dynamic filtering network for rgb-d salient object detection. In *ECCV*, 2020. 5, 7

[39] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Multi-scale interactive network for salient object detection. In *CVPR*, 2020. 5, 6

[40] Fulin Qi, Xin Tan, Zhizhong Zhang, Mingang Chen, Yuan Xie, and Lizhuang Ma. Glass makes blurs: Learning the visual blurriness for glass surface detection. *IEEE Transactions on Industrial Informatics*, 20(4):6631–6641, 2024. 4

[41] Olaf R., Philipp F., and Thomas B. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 6

[42] Daniel Seichter, Söhnke Fischedick, Mona Köhler, and Horst-Michael Gross. Efficient multi-task rgb-d scene analysis for indoor environments. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2022. 7

[43] Avishek Siris, Jianbo Jiao, Gary K.L. Tam, Xianghua Xie, and Rynson W.H. Lau. Scene context-aware salient object detection. In *ICCV*, 2021. 4

[44] Zhenbo Song, Zhenyuan Zhang, Kaihao Zhang, Wenhan Luo, Zhaoxin Fan, Wenqi Ren, and Jianfeng Lu. Robust single image reflection removal against adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24688–24698, 2023. 8

[45] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 4

[46] Tianyu Sun, Guodong Zhang, Wenming Yang, Jing-Hao Xue, and Guijin Wang. TROSD: A new RGB-D dataset for transparent and reflective object segmentation in practice. *IEEE TCSVT*, 2023. 1, 3, 5, 7, 8

[47] Xin Tan, Jiaying Lin, Ke Xu, Pan Chen, Lizhuang Ma, and Rynson W.H. Lau. Mirror detection with the visual chirality cue. *IEEE TPAMI*, 2023. 5, 6, 7

[48] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *IJCV*, 2019. 7

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[50] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020. 6

[51] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021. 1, 2, 7

[52] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):1–10, 2022. 1, 3, 6, 7

[53] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *CVPR*, 2023. 7

[54] Yu Wang, Quan Zhou, Jia Liu, Jian Xiong, Guangwei Gao, Xiaofu Wu, and Longin Jan Latecki. LEDNet: A lightweight encoder-decoder network for real-time semantic segmentation. In *ICIP*, 2019. 6

[55] Yonghui Wang, Wengang Zhou, Yunyao Mao, and Houqiang Li. Detect any shadow: Segment anything for video shadow detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(5):3782–3794, 2024. 6

[56] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *CVPR*, 2020. 5, 6

[57] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. In *CVPR*, 2019. 5, 6

[58] Kaite Xiang, Kailun Yang, and Kaiwei Wang. Polarization-driven semantic segmentation via efficient attention-bridged fusion. *Optica Express*, 2021. 4

[59] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. In *ECCV*, 2020. 4, 6, 7

[60] Enze Xie, Wenjia Wang, Wenhai Wang, Peize Sun, Hang Xu, Ding Liang, and Ping Luo. Segmenting transparent object in the wild with transformer. In *IJCAI*, 2021. 1, 3, 4, 6, 7, 8

[61] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, José Manuel Álvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 4

[62] Zhifeng Xie, Sen Wang, Qiucheng Yu, Xin Tan, and Yuan Xie. Csfwinformer: Cross-space-frequency window transformer for mirror detection. *IEEE Transactions on Image Processing*, 33:1853–1867, 2024. 6

[63] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. In *NeurIPS*, 2022. 6

[64] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. DenseASPP for semantic segmentation in street scenes. In *CVPR*, 2018. 6

[65] Xin Yang, Haiyang Mei, Ke Xu, Xiaopeng Wei, Baocai Yin, and R.W.H. Lau. Where is my mirror? In *ICCV*, 2019. 1, 3, 5, 6

[66] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 2018. 6

[67] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis E.H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, 2021. 4

[68] Yuhui Yuan, Lang Huang, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. OCNet: Object context for semantic segmentation. *IJCV*, 2021. 7

[69] Mingfeng Zha, Feiyang Fu, Yunqiang Pei, Guoqing Wang, Tianyu Li, Xiongxin Tang, Yang Yang, and Heng Tao Shen. Dual domain perception and progressive refinement for mirror detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 6, 7

[70] Mingfeng Zha, Yunqiang Pei, Guoqing Wang, Tianyu Li, Yang Yang, Wenbin Qian, and Heng Tao Shen. Weakly-supervised mirror detection via scribble annotations. In *Proceedings of the AAAI conference on artificial intelligence*, pages 6953–6961, 2024. 6, 7

[71] Jiaming Zhang, Kailun Yang, Angela Constantinescu, Kunyu Peng, Karin Müller, and Rainer Stiefelhagen. Trans4trans: Efficient transformer for transparent object and semantic scene segmentation in real-world navigation assistance. *IEEE T-ITS*, 2022. 6, 7

[72] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 4, 6, 7

[73] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. ICNet for real-time semantic segmentation on high-resolution images. In *ECCV*, 2018. 6

[74] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. PSANet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018. 4

[75] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 4

[76] Wujie Zhou, Enquan Yang, Jingsheng Lei, and Lu Yu. Frnet: Feature reconstruction network for rgb-d indoor scene parsing. *IEEE Journal of Selected Topics in Signal Processing*, 2022. 7

[77] Wujie Zhou, Yuqi Cai, Liting Zhang, Weiqing Yan, and Lu Yu. Utlnet: Uncertainty-aware transformer localization network for rgb-depth mirror segmentation. *IEEE Transactions on Multimedia*, 26:4564–4574, 2023. 6, 7

[78] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *ICCV*, 2019. 7