

Segmenting Transparent Object in the Wild with Transformer

Enze Xie^{1*}, Wenjia Wang², Wenhai Wang³, Peize Sun¹,
Hang Xu⁴, Ding Liang², Ping Luo¹

¹The University of Hong Kong

²Sensetime Retsearch

³Nanjing University

⁴Huawei Noah’s Ark Lab

Abstract

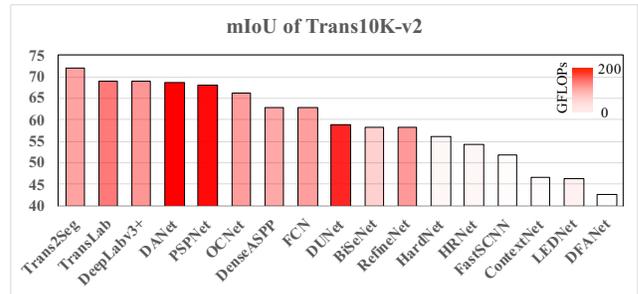
This work presents a new fine-grained transparent object segmentation dataset, termed Trans10K-v2, extending Trans10K-v1, the first large-scale transparent object segmentation dataset. Unlike Trans10K-v1 that only has two limited categories, our new dataset has several appealing benefits. (1) It has 11 fine-grained categories of transparent objects, commonly occurring in the human domestic environment, making it more practical for real-world application. (2) Trans10K-v2 brings more challenges for the current advanced segmentation methods than its former version. Furthermore, a novel Transformer-based segmentation pipeline termed Trans2Seg is proposed. Firstly, the Transformer encoder of Trans2Seg provides the global receptive field in contrast to CNN’s local receptive field, which shows excellent advantages over pure CNN architectures. Secondly, by formulating semantic segmentation as a problem of dictionary look-up, we design a set of learnable prototypes as the query of Trans2Seg’s Transformer decoder, where each prototype learns the statistics of one category in the whole dataset. We benchmark more than 20 recent semantic segmentation methods, demonstrating that Trans2Seg significantly outperforms all the CNN-based methods, showing the proposed algorithm’s potential ability to solve transparent object segmentation. Code is available in github.com/xieenze/Trans2Seg.

1 Introduction

Modern robots, mainly mobile robots and mechanical manipulators, would benefit a lot from the efficient perception of the transparent objects in residential environments since the environments vary drastically. The increasing utilization of glass wall and transparent door in the building interior and the glass cups and bottles in residential rooms has resulted in the wrong detection in various range sensors. In robotic research, most systems perceive the environment by multi-data sensor fusion via sonars or lidars. The sensors are relatively



(a) Selected images and corresponding high-quality masks.



(b) Performance comparison on Trans10K-v2.

Figure 1: (a) shows the high diversity of our dataset and high-quality annotations. (b) is **Comparisons** between Trans2Seg and other CNN-based semantic segmentation methods. All methods are trained on Trans10K-v2 with same epochs. mIoU is chosen as the metric. Deeper color bar indicates methods with larger FLOPS. Our Trans2Seg significantly surpasses other methods with lower flops.

consistent in detecting opaque objects but are still affected by the scan mismatching due to transparent objects. The unique feature of reflection, refraction, and light projection from the transparent objects may confuse the sensors. *Thus a reliable vision-based method, which is much cheaper and more robust than high-precision sensors, would be efficient.*

Although some transparent objects dataset [Chen *et al.*, 2018a; Mei *et al.*, 2020] were proposed, there are some obvious problems. (1) Limited dataset scale. These datasets often have less than 1K images captured from the real-world and less than 10 unique objects. (2) Poor diversity. The scene of these datasets is monotonous. (3) Fewer classes. All these datasets have only two classes, background and transparent objects. They lack fine-grained categories, which limited their practicality. Recently, [Xie *et al.*, 2020] proposed

*Contact Author

a large-scale and high-diversity dataset termed Trans10K, which divide transparent objects as ‘Things’ and ‘Stuff’. The dataset is high diversity, but it also lacks fine-grained transparent categories.

In this paper, we propose a fine-grained transparent object segmentation dataset termed Trans10K-v2 with more elaborately defined categories. The images are inherited from Trans10K-v1 [Xie *et al.*, 2020]. We annotate the 10428 images with 11 fine-grained categories: shelf, jar, freezer, window, glass door, eyeglass, cup, glass wall, glass bowl, water bottle, storage box. In Trans10K-v1, transparent **things** are defined to be grabbed by the manipulators and **stuff** are for robot navigation. Though two basic categories can partially help robots to interact with transparent objects, the provided fine-grained classes in Trans10K-v2 can provide more. We analyze these objects’ functions and how robots interact with them in appendix.

Based on this challenging dataset, we design Trans2Seg, introducing Transformer into segmentation pipeline for its encoder-decoder architecture. First, the Transformer encoder provides a global receptive field via self-attention. Larger receptive field is essential for segmenting transparent objects because transparent objects often share similar textures and context with its surroundings. Second, the decoder stacks successive layers to interact query embedding with Transformer encoder output. To facilitate the robustness of transparent objects, we carefully design a set of learnable class prototype embeddings as the query for Transformer decoder, and the key is the feature map from the Transformer encoder. Compared with convolutional paradigm, where the class prototypes is the fixed parameters of convolution kernel weight, our design provides a dynamic and context-aware implementation. As shown in Figure. 1b, we train and evaluate 20 existing representative segmentation methods on Trans10K-v2, and found that simply applying previous methods to this task is far from sufficient. By successfully introducing Transformer into this task, our Trans2Seg significantly surpasses the best TransLab [Xie *et al.*, 2020] by a large margin (72.1 vs. 69.0 on mIoU).

In summary, our main contributions are three-folds:

- We propose the largest glass segmentation dataset (Trans10K-v2) with 11 fine-grained glass image categories with diverse scenarios and high resolution. All the images are elaborately annotated with fine-shaped masks and function-oriented categories.
- We introduce a new Transformer-based network for transparent object segmentation with Transformer encoder-decoder architecture. Our method provides a global receptive field and is more dynamic in mask prediction, which shows excellent advantages.
- We evaluate more than 20 semantic segmentation methods on Trans10K-v2, and our Trans2Seg significantly outperforms these methods. Moreover, we show this task largely unsolved. Thus more research is needed.

2 Related Work

Semantic Segmentation. In deep learning era, convolutional neural network (CNN) puts forwards the develop-

ment of semantic segmentation in various datasets, such as ADE20K, CityScapes and PASCAL VOC. One of the pioneer works approaches, FCN [Long *et al.*, 2015], transfers semantic segmentation into an end-to-end fully convolutional classification network. For improving the performance, especially around object boundaries, [Chen *et al.*, 2017; Lin *et al.*, 2016] propose to use structured prediction module, conditional random fields (CRFs) [Chen *et al.*, 2014], to refine network output. Dramatic improvements in performance and inference speed have been driven by aggregating features at multiples scales, for example, PSPNet [Zhao *et al.*, 2017] and DeepLab [Chen *et al.*, 2017; Chen *et al.*, 2018b], and propagating structured information across intermediate CNN representations [Gadde *et al.*, 2016; Liu *et al.*, 2017; Wang *et al.*, 2018].

Transparent Object Datasets. [Chen *et al.*, 2018a] proposed TOM-Net. It contains 876 real images and 178K synthetic images which are generated by POV-Ray. However, only 4 unique objects are used in synthesizing the training data. Recently, [Xie *et al.*, 2020] introduce the first large-scale real-world transparent object segmentation dataset, termed Trans10K. It has 10K+ images while with only 2 categories. In this work, our Trans10K-v2 inherited the data and annotates 11 fine-grained categories.

Transformer in Vision Tasks. Transformer [Vaswani *et al.*, 2017] has been successfully applied in both high-level vision and low-level vision [Han *et al.*, 2020]. In ViT [Dosovitskiy *et al.*, 2020], Transformer is directly applied to sequences of image patches to complete image classification. In object detection areas [Carion *et al.*, 2020; Zhu *et al.*, 2020], DETR reasons about the relations of the object queries and the global image context via Transformer and outputs the final set of predictions in parallel without non-maximum suppression(NMS) procedures and anchor generation. SETR [Zheng *et al.*, 2020] views semantic segmentation from a sequence-to-sequence perspective with Transformer. IPT [Chen *et al.*, 2020] applies Transformer model to low-level computer vision task, such as denoising, super-resolution and deraining. In video processing, Transformer has received significantly growing attention. VisTR [Wang *et al.*, 2020] accomplishes instance sequence segmentation by Transformer. Multiple-object tracking [Sun *et al.*, 2020; Meinhardt *et al.*, 2021] employs Transformers to decode object queries and feature queries of the previous frame into bounding boxes of the current frame, and merged by Hungarian Algorithm or NMS.

3 Trans10K-v2 Dataset

Dataset Introduction. Our Trans10K-v2 dataset is based on Trans10K dataset [Xie *et al.*, 2020]. Following Trans10K, we use 5000, 1000 and 4428 images in training, validation and testing respectively. The distribution of the images is abundant in occlusion, spatial scales, perspective distortion. We further annotate the images with more fine-grained categories due to the functional usages of different objects. Trans10K-v2 dataset contains 10,428 images, with two main categories and 11 fine-grained categories: (1)



Figure 2: **Images in Trans10K-v2 dataset are carefully annotated with high quality.** The first row shows sample images and the second shows the segmentation masks. The color scheme which encodes the object categories are listed on the right of the figure. Zoom in for best view.

| Trans10Kv2 | shelf | door | wall | box | freezer | window | cup | bottle | jar | bowl | eyeglass |
|----------------|-------|------|-------|------|---------|--------|-------|--------|------|------|----------|
| image num | 280 | 1572 | 3059 | 603 | 90 | 501 | 3315 | 1472 | 997 | 340 | 410 |
| CMCC | 3.36 | 5.19 | 5.61 | 2.57 | 3.36 | 4.27 | 1.97 | 1.82 | 1.99 | 1.31 | 2.56 |
| pixel ratio(%) | 2.49 | 9.23 | 38.42 | 3.67 | 1.02 | 4.28 | 22.61 | 6.23 | 6.75 | 3.67 | 0.78 |

Table 1: **Statistic information of Trans10K-v2.** ‘CMCC’ denotes Mean Connected Components of each category. It is calculated by dividing the connected components number of a certain category by the image number. It represents the complexity of the transparent objects. ‘image num’ denotes the image number. ‘pixel ratio’ is the pixel number of a certain category accounts in all the pixels of transparent objects in Trans10K-v2.

Transparent **Things** contain **cup, bottle, jar, bowl** and **eyeglass**. (2) Transparent **Stuff** contain **windows, shelf, box, freezer, glass walls** and **glass doors**. In respect to fine-grained categories and high diversity, Trans10K-v2 is very challenging, and have promising potential in both computer vision and robotic researches.

Annotation Principle. The transparent objects are manually labeled by expert annotators with professional labeling tool. The annotators were asked to provide more than 100 points when they trace the boundaries of each transparent object, which ensures the high-quality outline of the mask shapes. The way of annotation is mostly the same with semantic segmentation datasets such as ADE20K. We set the background with 0, and the 11 categories from 1 to 11. We also provide the scene environment of each image locates at. The annotators are asked to strictly following principles when they label the images: (I) Highly transparent pixels of objects no matter made of glass, plastics or crystals are annotated as masks, other semi-transparent and non-transparent pixels are ignored. (II) When occluded by opaque objects, the pixels will be cropped from the masks. (III) The detailed principle of how we categorize the objects is listed in appendix.

Dataset Statistics. The statistic information of CMCC, image number, pixel proportion are listed in Table 1 in detail. From Table1, the sum of all the image numbers is larger than 10428 since some image has multiple categories of objects. See the caption for detail.

Evaluation Metrics. Results are reported in three metrics that are widely used in semantic segmentation to benchmark the performance of fine-grained transparent object segmentation. (1) **Pixel Accuracy** indicates the proportion of correctly classified pixels. (2) **Mean IoU** indicates mean intersection

over union. (3) **Category IoU** indicates the intersection over union of each category.

4 Method

4.1 Overall Pipeline

The overall Trans2Seg architecture contains a CNN backbone, an encoder-decoder Transformer, and a small convolutional head, as shown in Figure 3. For an input image of $(H, W, 3)$,

- The CNN backbone generates image feature map of $(\frac{H}{16}, \frac{W}{16}, C)$.
- The encoder takes in the summation of flattened feature of $(\frac{H}{16}, \frac{W}{16}, C)$ and positional embedding of $(\frac{H}{16}, \frac{W}{16}, C)$, and outputs encoded feature of $(\frac{H}{16}, \frac{W}{16}, C)$.
- The decoder interacts the learned class prototypes of (N, C) with encoded feature, and generates attention map of $(N, M, \frac{H}{16}, \frac{W}{16})$, where N is number of categories, M is number of heads in multi-head attention.
- The small convolutional head up-samples the attention map to $(N, M, \frac{H}{4}, \frac{W}{4})$, fuses it with high-resolution feature map Res2 and outputs attention map of $(N, \frac{H}{4}, \frac{W}{4})$.

The final segmentation is obtained by pixel-wise argmax operation on the output attention map.

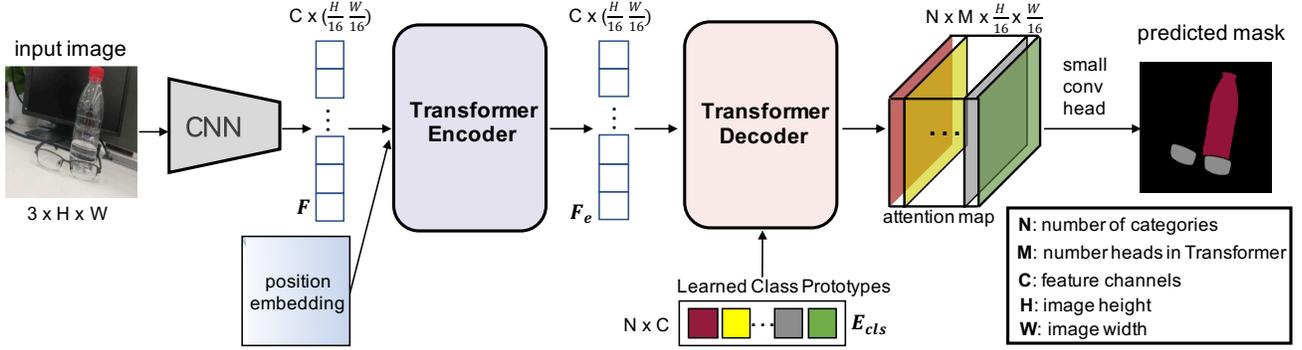


Figure 3: **The whole pipeline of our hybrid CNN-Transformer architecture.** First, the input image is fed to CNN to extract features F . Second, for Transformer encoder, the features and position embedding are flattened and fed to Transformer for self-attention, and output feature (F_e) from Transformer encoder. Third, for Transformer decoder, we specifically **define a set of learnable class prototype embeddings (E_{cls}) as query**, F_e as key, and calculate the attention map with E_{cls} and F_e . Each class prototype embedding corresponds to a category of final prediction. We also add a small conv head to fuse attention map and Res2 feature from CNN backbone. Details of Transformer decoder and small conv head refer to Figure 4. Finally, we can get the predict results by doing pixel-wise argmax on the attention map. For example, in this figure, the segmentation mask of two categories (**Bottle** and **Eyeglass**) corresponds to two class prototypes with same colors.

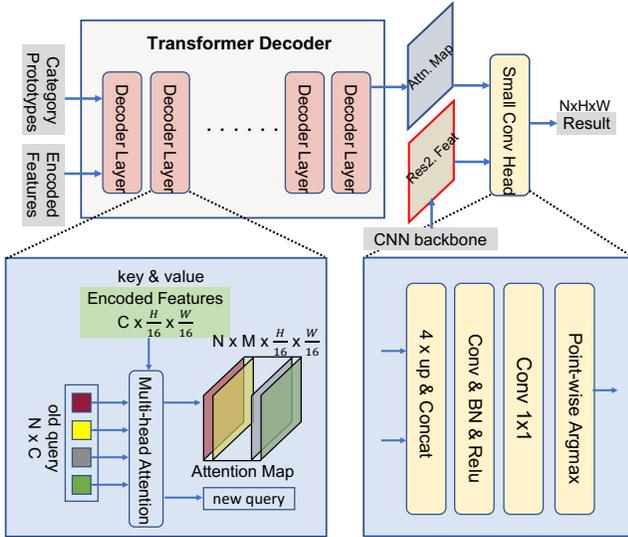


Figure 4: **Detail of Transformer Decoder and small conv head.** Input: The learnable category prototypes as query, features from Transformer encoder as key and value. The inputs are fed to Transformer decoder, which consists of several decoder layers. The attention map from last decoder layer and the Res2 feature from CNN backbone are combined and fed to a small conv head to get final prediction result.

4.2 Encoder

The Transformer encoder takes a sequence as input, so the spatial dimensions of the feature map $(\frac{H}{16}, \frac{W}{16}, C)$ is flattened into one dimension $(\frac{H}{16} \frac{W}{16}, C)$. To compensate missing spatial dimensions, positional embedding [Gehring *et al.*, 2017] is supplemented to one dimension feature to provide information about the relative or absolute position of the feature in the sequence. The positional embedding has the same dimension $(\frac{H}{16} \frac{W}{16}, C)$ with the flattened feature. The encoder is composed of stacked encoder layers, each of which con-

sists of a multi-head self-attention module and a feed forward network [Vaswani *et al.*, 2017].

4.3 Decoder

The Transformer decoder takes input a set of learnable class prototype embeddings as query, denoted by E_{cls} , the encoded feature as key and value, denoted by F_e , and output the attention map followed by Small Conv Head to obtain final segmentation result, as shown in Figure 4.

The class prototype embeddings are learned category prototypes, updated iteratively by a series of decoder layers through multi-head attention mechanisms. We denoted iterative update rule by \odot , then the class prototype in each decoder layer is:

$$E_{cls}^s = \odot_{i=0, \dots, s-1} \text{softmax}(E_{cls}^i F_e) F_e \quad (1)$$

In the final decoder layer, the attention map is extracted out to into small conv head:

$$\text{attention map} = E_{cls}^s F_e \quad (2)$$

The pseudo code of small conv head is shown in shown in Figure 4. The attention map from Transformer decode is the shape of $(N, M, \frac{H}{16} \frac{W}{16})$, where N is number of categories, M is number of heads in multi-head attention. It is up-sampled to $(N, M, \frac{H}{4}, \frac{W}{4})$, then fused with high-resolution feature map Res2 in the second dimension to $(N, M+C, \frac{H}{4}, \frac{W}{4})$, and finally transformed into output attention map of $(N, \frac{H}{4}, \frac{W}{4})$. The final segmentation is obtained by pixel-wise argmax operation on the output attention map.

4.4 Discussion

The most related work with Trans2Seg is SETR and DETR [Zheng *et al.*, 2020; Carion *et al.*, 2020]. In this section we discuss the relations and differences in details.

SETR . Trans2Seg and SETR are both segmentation pipelines. Their key difference is reflected in the design of the decoder. In SETR, the decoder is simple several convolutional layers, which is similar with most previous methods. However, the decoder of Trans2Seg is also Transformer, which fully utilize the advantages of attention mechanism in semantic segmentation.

DETR . Trans2Seg and DETR share similar components in the pipeline, including CNN backbone, Transformer encoder and decoder. The biggest difference is the definition of query. In DETR, the decoder’s queries represents N learnable objects because DETR is designed for object detection. However, in Trans2Seg, the queries represents N learnable class prototypes, where each query represents one category. We could see that the minor change on query design could generalize Transformer architecture to apply to diverse vision tasks, such as object detection and semantic segmentation.

5 Experiments

5.1 Implementation Details.

We implement Trans2Seg with Pytorch. The ResNet-50 [He *et al.*, 2016] with dilation convolution at last stage is adopted as the CNN extractor. For loss optimization, we use Adam optimizer with epsilon $1e-8$ and weight decay $1e-4$. Batch size is 8 per GPU. We set learning rate $1e-4$ and decayed by the poly strategy [Yu *et al.*, 2018] for 50 epochs. We use 8 V100 GPUs for all experiments. For all CNN based methods, we random scale and crop the image to 480×480 in training, and resize image to 513×513 in inference, following common setting on PASCAL VOC [Everingham and Winn, 2011]. For our Trans2Seg, we adopt Transformer architecture and need to keep the shape of learned position embedding same in training/inference, so we directly resize the image to 512×512 . Code has been released for community to follow.

5.2 Ablation Studies.

We use the FCN [Long *et al.*, 2015] as our baseline. FCN is a fully convolutional network with very simple design, and it is also a very classic semantic segmentation method. First, we demonstrate that Transformer encoder can build long range attention between pixels, which has much larger receptive field than CNN filters. Second, we remove the CNN decoder in FCN and replace by our Transformer decoder, we design a set of learnable class prototypes as queries and show that this design further helps improve the accuracy. Third, we verify our method with Transformer at different scales.

Self-Attention of Transformer Encoder. As shown in Table 2, the FCN baseline without Transformer encoder achieves 62.7% mIoU. When adding Transformer encoder, the mIoU directly improves 6.1%, achieving 66.8% mIoU. It demonstrates that the self-attention module in Transformer encoder provides global receptive field, which is better than CNN’s local receptive field in transparent object segmentation.

Category Prototypes of Transformer Decoder. In Table 2, we verify the effectiveness of learnable category prototypes in Transformer decoder. In row 2, with traditional CNN

| id | Trans. Enc. | Trans. Dec. | CNN Dec. | mIoU |
|----|-------------|-------------|----------|------|
| 0 | × | × | ✓ | 62.7 |
| 1 | ✓ | × | ✓ | 68.8 |
| 2 | ✓ | ✓ | × | 72.1 |

Table 2: **Effectiveness of Transformer encoder and decoder.** ‘Trans.’ indicates Transformer. ‘Enc.’ and ‘Dec.’ means encoder and decoder.

| Scale | hyper-param. | GFlops | MParams | mIoU |
|--------|--------------|--------|---------|------|
| small | e128-n1-m2 | 40.9 | 30.5 | 69.2 |
| medium | e256-n4-m3 | 49.0 | 56.2 | 72.1 |
| large | e768-n12-m4 | 221.8 | 327.5 | 70.3 |

Table 3: **Performance of Transformer at different scales.** ‘e{a}-n{b}-m{c}’ means the Transformer with number of ‘a’ embedding dims, ‘b’ layers and ‘c’ mlp ratio.

decoder, the mIoU is 68.8%. However, with our Transformer decoder, the mIoU boosts up to 72.1% with 3.3% improvement. The strong performance benefits from the flexible representation that learnable category prototypes as queries to find corresponding pixels in feature map.

Scale of Transformer. The scale of Transformer is mainly influenced by three hyper-parameters: (1) Embedding dim of feature. (2) Number of attention layers. (3) MLP ratio in feed forward layer. We are interested in whether enlarging the model size can continuously improve performance. So we set three combinations, as shown in Figure 3. We can find that with the increase of the size of Transformer, the mIoU first increases then decreases. We argue that without massive pre-trained data, *e.g.* the large-scale nlp data BERT [Devlin *et al.*, 2019] used, the size of Transformer is not the larger the better for our task.

5.3 Comparison to the state-of-the-art.

We select more than 20 semantic segmentation methods: Translab, Deeplabv3+, DABNet, PSPNet, OCNet, DenseASpp, FCN, UNet, BiseNet, RefineNet, HardNet, HRNet, FastSCNN, ContextNet, LedNet, DUNet, ICNet, FPENet, DFANet, DANet, ESPNetv2, to evaluate on our Trans10K-v2 dataset, the methods selection largely follows the benchmark of TransLab. For fair comparison, we train all the methods with 50 epochs.

Table 4 reports the overall quantitative comparison results on test set. Our Trans2Seg achieves state-of-the-art 72.15% mIoU and 94.14% pixel ACC, significant outperforms other pure CNN-based methods. For example, our method is 2.1% higher than TransLab, which is the previous SOTA method. We also find that our method tend to performs much better on small objects, such as ‘bottle’ and ‘eyeglass’ (10.0% and 5.0% higher than previous SOTA). We consider that the Transformer’s long range attention benefits the small transparent object segmentation.

In Figure 5, we visualize the mask prediction of Trans2Seg and other CNN-based methods. We can find that owing to Transformer’s large receptive field and attention mechanism, our method can distinguish background and different categories transparent objects much better than other methods,

| Method | FLOPs | ACC \uparrow | mIoU \uparrow | Category IoU \uparrow | | | | | | | | | | | |
|------------------|--------|----------------|-----------------|-------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | | | bg | shelf | Jar | freezer | window | door | eyeglass | cup | wall | bowl | bottle | box |
| FPENet | 0.76 | 70.31 | 10.14 | 74.97 | 0.01 | 0.00 | 0.02 | 2.11 | 2.83 | 0.00 | 16.84 | 24.81 | 0.00 | 0.04 | 0.00 |
| ESPNetv2 | 0.83 | 73.03 | 12.27 | 78.98 | 0.00 | 0.00 | 0.00 | 0.00 | 6.17 | 0.00 | 30.65 | 37.03 | 0.00 | 0.00 | 0.00 |
| ContextNet | 0.87 | 86.75 | 46.69 | 89.86 | 23.22 | 34.88 | 32.34 | 44.24 | 42.25 | 50.36 | 65.23 | 60.00 | 43.88 | 53.81 | 20.17 |
| FastSCNN | 1.01 | 88.05 | 51.93 | 90.64 | 32.76 | 41.12 | 47.28 | 47.47 | 44.64 | 48.99 | 67.88 | 63.80 | 55.08 | 58.86 | 24.65 |
| DFANet | 1.02 | 85.15 | 42.54 | 88.49 | 26.65 | 27.84 | 28.94 | 46.27 | 39.47 | 33.06 | 58.87 | 59.45 | 43.22 | 44.87 | 13.37 |
| ENet | 2.09 | 71.67 | 8.50 | 79.74 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 22.25 | 0.00 | 0.00 | 0.00 |
| HRNet_w18 | 4.20 | 89.58 | 54.25 | 92.47 | 27.66 | 45.08 | 40.53 | 45.66 | 45.00 | 68.05 | 73.24 | 64.86 | 52.85 | 62.52 | 33.02 |
| HardNet | 4.42 | 90.19 | 56.19 | 92.87 | 34.62 | 47.50 | 42.40 | 49.78 | 49.19 | 62.33 | 72.93 | 68.32 | 58.14 | 65.33 | 30.90 |
| DABNet | 5.18 | 77.43 | 15.27 | 81.19 | 0.00 | 0.09 | 0.00 | 4.10 | 10.49 | 0.00 | 36.18 | 42.83 | 0.00 | 8.30 | 0.00 |
| LEDNet | 6.23 | 86.07 | 46.40 | 88.59 | 28.13 | 36.72 | 32.45 | 43.77 | 38.55 | 41.51 | 64.19 | 60.05 | 42.40 | 53.12 | 27.29 |
| ICNet | 10.64 | 78.23 | 23.39 | 83.29 | 2.96 | 4.91 | 9.33 | 19.24 | 15.35 | 24.11 | 44.54 | 41.49 | 7.58 | 27.47 | 3.80 |
| BiSeNet | 19.91 | 89.13 | 58.40 | 90.12 | 39.54 | 53.71 | 50.90 | 46.95 | 44.68 | 64.32 | 72.86 | 63.57 | 61.38 | 67.88 | 44.85 |
| DenseASPP | 36.20 | 90.86 | 63.01 | 91.39 | 42.41 | 60.93 | 64.75 | 48.97 | 51.40 | 65.72 | 75.64 | 67.93 | 67.03 | 70.26 | 49.64 |
| DeepLabv3+ | 37.98 | 92.75 | 68.87 | 93.82 | 51.29 | 64.65 | 65.71 | 55.26 | 57.19 | 77.06 | 81.89 | 72.64 | 70.81 | 77.44 | 58.63 |
| FCN | 42.23 | 91.65 | 62.75 | 93.62 | 38.84 | 56.05 | 58.76 | 46.91 | 50.74 | 82.56 | 78.71 | 68.78 | 57.87 | 73.66 | 46.54 |
| OCNet | 43.31 | 92.03 | 66.31 | 93.12 | 41.47 | 63.54 | 60.05 | 54.10 | 51.01 | 79.57 | 81.95 | 69.40 | 68.44 | 78.41 | 54.65 |
| RefineNet | 44.56 | 87.99 | 58.18 | 90.63 | 30.62 | 53.17 | 55.95 | 42.72 | 46.59 | 70.85 | 76.01 | 62.91 | 57.05 | 70.34 | 41.32 |
| Translab | 61.31 | 92.67 | 69.00 | 93.90 | 54.36 | 64.48 | 65.14 | 54.58 | 57.72 | 79.85 | 81.61 | 72.82 | 69.63 | 77.50 | 56.43 |
| DUNet | 123.69 | 90.67 | 59.01 | 93.07 | 34.20 | 50.95 | 54.96 | 43.19 | 45.05 | 79.80 | 76.07 | 65.29 | 54.33 | 68.57 | 42.64 |
| UNet | 124.55 | 81.90 | 29.23 | 86.34 | 8.76 | 15.18 | 19.02 | 27.13 | 24.73 | 17.26 | 53.40 | 47.36 | 11.97 | 37.79 | 1.77 |
| DANet | 198.00 | 92.70 | 68.81 | 93.69 | 47.69 | 66.05 | 70.18 | 53.01 | 56.15 | 77.73 | 82.89 | 72.24 | 72.18 | 77.87 | 56.06 |
| PSPNet | 187.03 | 92.47 | 68.23 | 93.62 | 50.33 | 64.24 | 70.19 | 51.51 | 55.27 | 79.27 | 81.93 | 71.95 | 68.91 | 77.13 | 54.43 |
| Trans2Seg | 49.03 | 94.14 | 72.15 | 95.35 | 53.43 | 67.82 | 64.20 | 59.64 | 60.56 | 88.52 | 86.67 | 75.99 | 73.98 | 82.43 | 57.17 |

Table 4: **Evaluated state-of-the-art semantic segmentation methods.** Sorted by FLOPs. Our proposes Trans2Seg surpasses all the other methods in pixel accuracy and mean IoU, as well as most of the category IoUs (8 in 11).

especially when the image contains multiple objects of different categories. Moreover, our method can obtain high quality detail information, *e.g.* boundary of object, and tiny transparent objects, while other CNN-based methods fail to do so. More results are shown in supplementary material.

6 Conclusion

In this paper, we present a challenging fine-grained transparent object segmentation dataset with 11 common categories, termed Trans10K-v2. Moreover, we propose a Transformer-based pipeline with encoder having global receptive field and decoder with category query, termed Trans2Seg, to solve this challenging task. Finally, we evaluate more than 20 mainstream semantic segmentation methods and shows that our Trans2Seg clearly surpass these CNN-based segmentation methods.

In the future, we are interested in exploring Transformer encoder-decoder on general segmentation tasks, such as Cityscapes and PASCAL VOC. We will also put more efforts to solve transparent object segmentation task.

Acknowledgements

This work is partially supported by the General Research Fund of Hong Kong No. 27208720, and the Research Donation from Huawei. We Thank Yaojun Liu for insightful discussion.

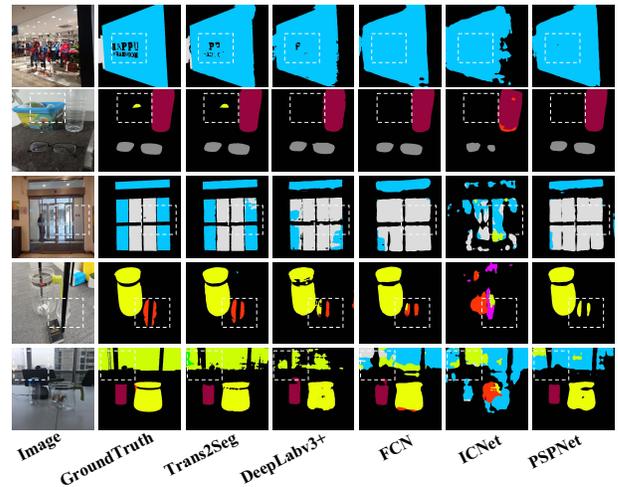


Figure 5: **Visual comparison** of Trans2Seg to other CNN-based semantic segmentation methods. Our Trans2Seg clearly outperforms others thanks to the Transformer’s global receptive field and attention mechanism, especially in dash region. Zoom in for best view. Refer to supplementary materials for more visualized results.

References

- [Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End object detection with transformers. In *ECCV*, 2020.
- [Chen *et al.*, 2014] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. 2014.
- [Chen *et al.*, 2017] 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. *TPAMI*, 2017.
- [Chen *et al.*, 2018a] Guanying Chen, Kai Han, and Kwan-Yee K. Wong. Tom-net: Learning transparent object matting from a single image. In *CVPR*, 2018.
- [Chen *et al.*, 2018b] 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.
- [Chen *et al.*, 2020] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv*, 2020.
- [Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *NAACL-HLT 2019*, 2019.
- [Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, 2020.
- [Everingham and Winn, 2011] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep.*, 2011.
- [Gadde *et al.*, 2016] Raghudeep Gadde, Varun Jampani, Martin Kiefel, Daniel Kappler, and Peter V Gehler. Superpixel convolutional networks using bilateral inceptions. In *ECCV*, 2016.
- [Gehring *et al.*, 2017] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. *arXiv*, 2017.
- [Han *et al.*, 2020] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on visual transformer. *arXiv*, 2020.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [Lin *et al.*, 2016] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016.
- [Liu *et al.*, 2017] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NIPS*, 2017.
- [Long *et al.*, 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.
- [Mei *et al.*, 2020] 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*, June 2020.
- [Meinhardt *et al.*, 2021] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackerformer: Multi-object tracking with transformers. *arXiv*, 2021.
- [Sun *et al.*, 2020] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv*, 2020.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeuralIPS*, 2017.
- [Wang *et al.*, 2018] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.
- [Wang *et al.*, 2020] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv*, 2020.
- [Xie *et al.*, 2020] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. 2020.
- [Yu *et al.*, 2018] 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.
- [Zhao *et al.*, 2017] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.
- [Zheng *et al.*, 2020] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv*, 2020.
- [Zhu *et al.*, 2020] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv*, 2020.