Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.056093

ARTICLE

EGSNet: An Efficient Glass Segmentation Network Based on Multi-Level Heterogeneous Architecture and Boundary Awareness

Guojun Chen*, Tao Cui, Yongjie Hou and Huihui Li

Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, 266580, China

*Corresponding Author: Guojun Chen. Email: chengj@upc.edu.cn

Received: 14 July 2024 Accepted: 23 October 2024 Published: 19 December 2024

ABSTRACT

Existing glass segmentation networks have high computational complexity and large memory occupation, leading to high hardware requirements and time overheads for model inference, which is not conducive to efficiency-seeking real-time tasks such as autonomous driving. The inefficiency of the models is mainly due to employing homogeneous modules to process features of different layers. These modules require computationally intensive convolutions and weight calculation branches with numerous parameters to accommodate the differences in information across layers. We propose an efficient glass segmentation network (EGSNet) based on multi-level heterogeneous architecture and boundary awareness to balance the model performance and efficiency. EGSNet divides the feature layers from different stages into low-level understanding, semantic-level understanding, and global understanding with boundary guidance. Based on the information differences among the different layers, we further propose the multi-angle collaborative enhancement (MCE) module, which extracts the detailed information from shallow features, and the large-scale contextual feature extraction (LCFE) module to understand semantic logic through deep features. The models are trained and evaluated on the glass segmentation datasets HSO (Home-Scene-Oriented) and Trans10k-stuff, respectively, and EGSNet achieves the best efficiency and performance compared to advanced methods. In the HSO test set results, the IoU, F_β, MAE (Mean Absolute Error), and BER (Balance Error Rate) of EGSNet are 0.804, 0.847, 0.084, and 0.085, and the GFLOPs (Giga Floating Point Operations Per Second) are only 27.15. Experimental results show that EGSNet significantly improves the efficiency of the glass segmentation task with better performance.

KEYWORDS

Image segmentation; multi-level heterogeneous architecture; feature differences

1 Introduction

Glass materials are widely used in architectural designs. Due to the characteristics of the material and the influence of light on glass, the surface area of glass tends to blend in with the surroundings, which may pose a severe challenge to vision systems [1]. For example, in the glass regions, the accuracy of the object detection task [2] and the overlap degree of the 3D scene completion task [3] are both over 50% lower than the average value. The segmentation research of small glass items [4–7], such as glass

bottles and cups, has achieved breakthrough progress. While large-scale glass areas, due to the lack of fixed patterns and textures, such glass areas are prone to blend in with any background [8], which makes glass surface segmentation fundamentally different from other segmentation tasks. Moreover, the visual characteristics of glass exhibit differences in various weather conditions, making extracting the glass pixel features from the image challenging.

Glass surface segmentation refers to segmenting the glass surface pixels from the image to achieve a binary classification between the glass surface and the background area [1]. With the rapid development of deep learning, some scholars segment the glass surface through neural networks. Mei et al. [1] proposed the glass detection network (GDNet), which extracts context information through spatially separable convolution and fuses advanced semantic features with low-level features to improve the detection ability of large-scale glass areas; Xie et al. [7] proposed a transparent area extraction network based on Transformer, inputting the features extracted by the convolutional neural network into the Transformer encoder-decoder structure to enhance the global receptive field of the model; Lin et al. [9] proposed the GlassNet based on reflection detection, integrating the reflection information of the glass surface into the features to further locate the position of the glass surface; Liu et al. [10] proposed the VGSD network integrating the reflection of the glass surface position and the context reflection, fusing the two reflection features into the RGB features to improve the integrity of glass extraction, but this network has the phenomenon of false detection of other objects with reflective properties; Yu et al. [11] proposed the progressive segmentation network PGSNet, adopting a multi-scale encoder-decoder structure and cross-fusing different scale features to enhance the model's segmentation ability for different sizes of glass; Mei et al. [12] proposed a glass surface detection network based on intensity and spectral polarization cues, encoding and fusing the polarization angle and polarization intensity information into the RGB image features through images taken by a polarizing camera, but the three backbones are too heavy; Huo et al. [13] proposed a glass surface segmentation method based on thermal images, encoding the RGB and thermal images respectively and cross-fusing the features at the decoding stage; He et al. [14] proposed an edge-aware point-wised graph convolution network, optimizing the glass boundary extracted by the network through modeling the shape of the glass object, but there is a phenomenon of discontinuous pixels inside the glass; Lin et al. [15] proposed a glass surface extraction network based on semantic information, enhancing the network's attention to the glass area through the semantic feature encoding branch, but this method requires separate annotation of semantic information maps; Zhang et al. [16] proposed a detail-guided cross-level fusion network DCNet, which interacted multi-scale detail features to extract finer detail clues and used correlation and discontinuity to refine the glass boundary. Qi et al. [17] found that glass causes image blurriness and proposed a visual blurriness aggregation (VBA) module and a visual blurriness driven refinement (VBDR) module based on blurring cues to improve the glass detection accuracy. Tan et al. [18] proposed a glass detection method based on visual distortion, guiding the backbone to locate the glass initially through image distortion caused by glass and refining the mask progressively based on the amount of glass. Zheng et al. [19] proposed a glass segmentation network via mistake correction, correcting the errors in the identification stage through the gained experience of the correction stage. Qin et al. [20] guided the fine-grained boundary segmentation through the original semantic information and fused the features through a learnable Fourier convolutional controller. According to the above glass segmentation methods, based on the different inputs of the network, glass segmentation methods can be divided into the RGB method and the combination method, among which the combination methods require RGB and non-RGB image inputs to multiple encoders and complex methods to fuse features, which require significant computational overhead and parameters. Despite being more challenging, the RGB methods offer the advantages of low cost,

strong universality, and better model efficiency. In this paper, RGB images are used as the input of the glass segmentation network. The existing glass segmentation methods pursue segmentation accuracy but need better efficiency. In some scenes with more complex backgrounds, the existing methods are not effective enough to segment the glass boundary, and the problem of discontinuous pixels on the glass surface segmentation also occurs.

The main contributions of this paper are as follows:

(1) We design an efficient glass segmentation network through multi-level heterogeneous architecture and boundary awareness. The model improves the accuracy of glass surface segmentation and glass boundary extraction with less computational consumption and parameters.
(2) A multi-angle collaborative enhancement module is proposed to discover positional relationships and long-distance dependencies in feature maps from different angles and enhance the representation of low-level features by capturing more detailed information based on the contrast differences.
(3) A large-scale contextual feature extraction module is proposed to extract critical contexts at different scales and explore contextual differences between glass and its surroundings.

2 Related Work

In this section, we briefly review the segmentation methods related to the proposed method, including the methods for transparent object segmentation and mirror segmentation.

2.1 Transparent Object Segmentation

Transparent objects lack significant color and texture features, and their segmentation task has attracted attention in recent years. Chen et al. [4] proposed a two-stage transparent object extraction network, which is divided into a coarse-level extraction stage and a refinement extraction stage. In the coarse-level stage, VGG16 is used to extract image features, and after upsampling through three different branches, the coarse-grained mask, attenuation mask, and refraction flow field are obtained. The fusion result is inputted into the residual network [21] to get the segmentation result. Xie et al. [6] created a large-scale transparent object segmentation dataset for real-scene shooting and proposed a boundary-aware transparent object segmentation method called TransLab. TransLab consists of regular streams and boundary streams. The regular streams are used for transparent object segmentation, the boundary streams are used for target boundary prediction, and the advanced feature encoding in the middle part uses the Atrous Spatial Pyramid Pooling (ASPP) [22] module to enhance the network's receptive field. The boundary stream features are mapped to the regular stream in the form of attention, improving the network's ability to perceive boundaries. Cao et al. [23] noticed that boundary-based methods may mistake other boundaries for glass boundaries and detect the area surrounded by this boundary as a transparent object. Therefore, they proposed the FakeMix data enhancement strategy to alleviate the problem of imbalanced boundary data. In addition, they proposed an Adaptive Atrous Spatial Pyramid Pooling (AdaptiveASPP) module for adaptive boundary flow and regular flow features. Recently, Vision Transformer (ViT) has performed well in visual tasks such as detection, tracking, and segmentation [24–26]. Xie et al. [7] created a transparent object segmentation dataset containing 12 categories and proposed a Transformer-based transparent object extraction method called Trans2Seg. This network uses a Convolutional Neural Network (CNN) to extract image features and inputs the features into the Transformer encoder and decoder. The decoder adopts multi-level decoding layers and incorporates a multi-head attention mechanism. Finally, pixel classification is performed through a simple convolutional head. However,

due to adopting the CNN-Transformer architecture, Trans2Seg improves performance while reducing efficiency. Compared to other methods, boundary-aware methods extract and analyze boundary regions to segment boundaries more accurately and guide regular streams to locate target regions when transparent objects are hard to distinguish from the surroundings. In this paper, we extract the glass boundaries at two feature layers to refine the boundary features, guide the network to locate the glass surface, and supervise the boundary feature extraction effect of the complete feature map.

2.2 Mirror Segmentation

The mirror image is similar to the real scene, which increases the difficulty of mirror segmentation. Yang et al. [27] created a large-scale mirror detection dataset and proposed a mirror segmentation network based on multi-scale context contrast. This network explores the differences in context information inside and outside the mirror through the Context Contrast Feature Extraction (CCFE) module at different layers to segment the mirror surface. If the mirror image is similar to the surrounding environment, it is difficult for the CCFE module to detect the context contrast differences. Lin et al. [28] proposed a progressive mirror segmentation method, which extracts relationship context comparison through the Relative Context Contrasted Local (RCCL) module and obtains segmentation results at different scales through multi-level decoders. Finally, the refinement module fuses the extraction results with the boundary results at different scales. Huang et al. [29] proposed a symmetry-aware transformer network that takes the image and its mirror as dual-path inputs. The network perceives the symmetric relationships in the image through the symmetry-aware attention module (SAAM), discovers the contrastive contextual relationships and fuses feature differences through the contrast and fusion decoder module (CFDM). Although this network achieves significant performance improvements, its efficiency is compromised due to the adoption of a dual-branch transformer structure. To improve the efficiency of mirror detection, He et al. [30] proposed a multi-level heterogeneous network (HetNet). HetNet adopts the multi-orientation intensity-based contrasted (MIC) module for the low-level features to discover the contrast differences from different directions and adopts the reflection semantic logical (RSL) module for the high-level features. The RSL module uses large kernel convolutions to extract reflection semantic features of different scales. Due to the differentiated processing of different layers, this network greatly improves performance. Although mirrors and glasses are made of similar materials, it isn't easy to detect glasses through reflection information and the contrast differences between mirror images. In this paper, we adopt a multi-level heterogeneous architecture to improve network efficiency.

3 Proposed Method

To address the low efficiency of existing glass segmentation methods, as well as the issues of discontinuous surface pixels and rough boundaries in the segmentation results, based on the multi-level heterogeneous architecture of HetNet [30] and the boundary awareness of TransLab [6], combined with the visual characteristics of the glass surface, this paper proposes an Efficient Glass Segmentation Network (EGSNet) and further proposes a Multi-angle Collaborative Enhancement (MCE) module and a Large-scale Contextual Feature Extraction (LCFE) module for the various in the feature maps extracted from different layers. The overview of the network is shown in Fig. 1.

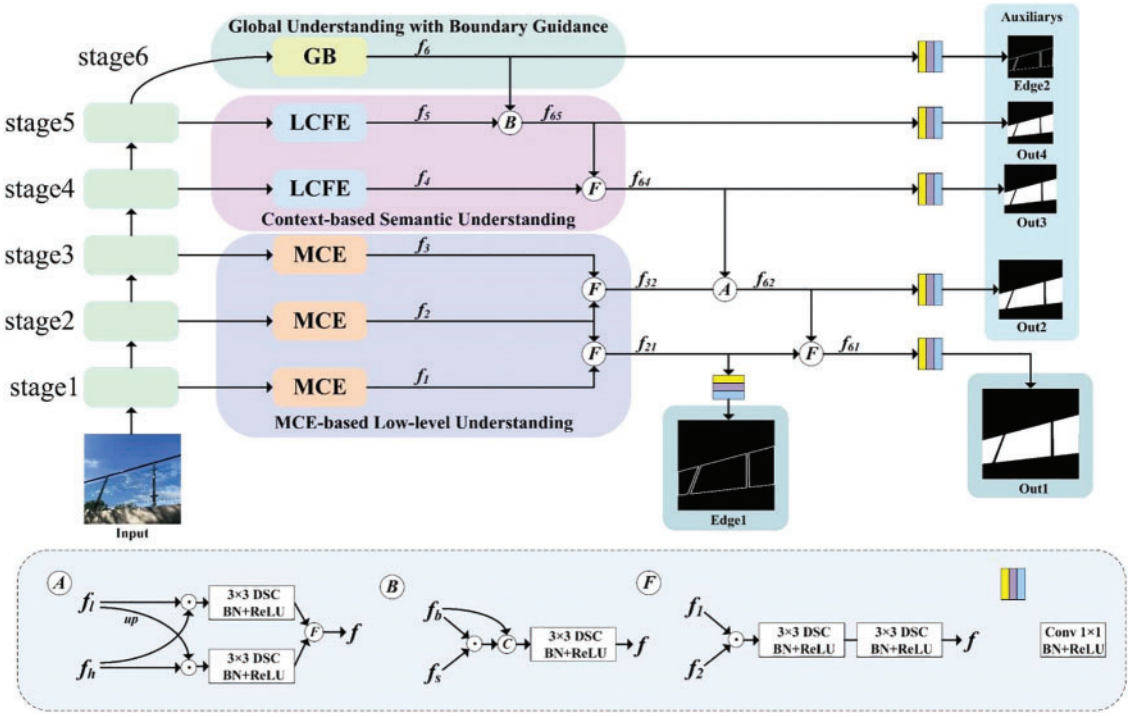

Figure 1: Efficient glass segmentation network structure

After extracting five scales of feature maps from ResNeXt-101 [31], we apply MCE modules for the low-level understanding of the image in the first three stages, capturing the details of the glass surface and sensitive positional information from different perspectives. For the deep features in stages 4 and 5, we apply LCFE modules for the semantic deep understanding of the features, exploring the contextual differences between the glass surface and the background. In the global understanding with boundary guidance (GB) of stage 6, we follow the Spatial Pyramid Pooling Extended Layer Aggregation Network (SPPELAN) [32] module to efficiently aggregate the global context in the deep features and guide the network to perceive the glass boundary features. The outputs from stages 1 to 6 are denoted as f_1–f_6, and different methods are employed to fuse the features. 3 × 3 depth-wise separable convolution (3 × 3 DSC) [33] is used for convolution operation in feature fusion to reduce the computational cost and the number of parameters. Fusion method *B* consists of element-wise product, channel-wise concatenation, 3 × 3 DSC, BN and ReLU; fusion method *F* comprises element-wise product, 3 × 3 DSC, BN and ReLU; and fusion method *A* is composed of upsampling, element-wise product, 3 × 3 DSC, BN and ReLU, and *F*. After fusing the features from different layers, simple heads composed of 1 × 1 convolution, BN and ReLU is utilized to obtain the results Out1 and Edge1, along with the auxiliary results Out2, Out3, Out4, and Edge2. The auxiliary outputs are only generated during the training process.

We follow Spatial Pyramid Pooling Extended Layer Aggregation Network (SPPELAN) [32] in stage 6, which employs Spatial Pyramid Pooling (SPP) to extract features at different scales and then applies Efficient Local Aggregation Network (ELAN) to aggregate these features, thus enhancing the multi-scale feature extraction and feature aggregation ability in this stage. SPPELAN is shown in Fig. 2. Compared with SPP, SPPELAN effectively extracts global information with less computational consumption. Therefore, using SPPELAN in global understanding with boundary guidance (GB) can

capture global information, extract finer glass boundaries, and guide other layers to localize the glass surface based on the boundary.

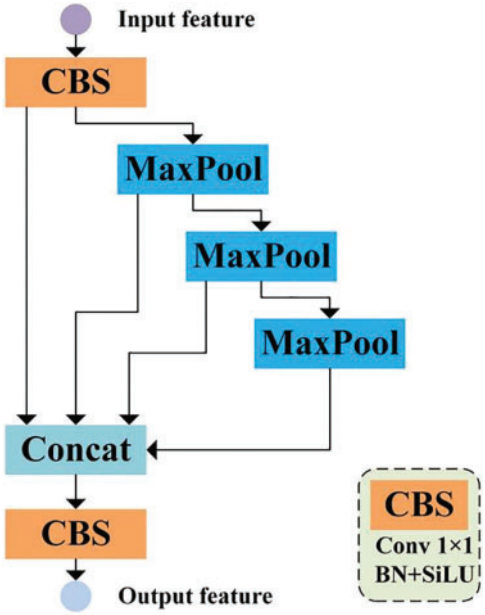

Figure 2: Spatial pyramid pooling extended layer aggregation network [32]

3.1 MCE Module

Low-level features contain detailed information, which is the key to extracting the details of the glass surface and the basis for constructing deep semantic features and accurately understanding the context relationship. Better capturing the coordinate position information in low-level features helps the network focus on important pixels and improves the accuracy of glass surface positioning. Coordinate Attention [34] employs 1×1 convolution to encode 1D features in the horizontal and vertical directions, captures long-range dependencies along the horizontal and vertical directions, and retains precise positional information, enhancing the network's perception ability of directions and positions. Coordinate Attention enhances the expression of essential channels and positions in the feature map with a relatively small computational cost [35], but the glass surface lacks visual characteristics. Therefore, performing attention enhancement on the feature map only from a single angle cannot reach the requirements of the glass segmentation task.

We propose the MCE module based on Coordinate Attention, which focuses on the feature differences in three angles to enhance low-level features, discovering detail cues such as blurring [17] and texture distortion [18], thereby locating the pixel position of the glass surface. As shown in Fig. 3, the MCE module feeds the original feature map, the feature map rotated 90° clockwise, and the feature map rotated 180° clockwise into three parallel Coordinate Attention blocks to capture precise positional information and long-range dependencies in three angles. Then, the feature maps are rotated back to the original angle, and the contrast differences of the three angles are calculated by element-wise product, 3×3 group convolution, BN and ReLU. Regarding performance, the MCE module weights the feature maps in terms of direction and position from three angles to enhance the representation of low-level features, and based on the contrast differences of features from different

angles, it can better discover the blurring, distortion and other detail clues caused by the glass surface, and thus locate the pixel positions of the glass surface with greater precision. Regarding efficiency, the MCE module adopts 1D feature encodings to calculate the attention weights and group convolution to find contrast differences, which cause a relatively small number of parameters and computational cost overhead. Therefore, the MCE module balances performance and efficiency.

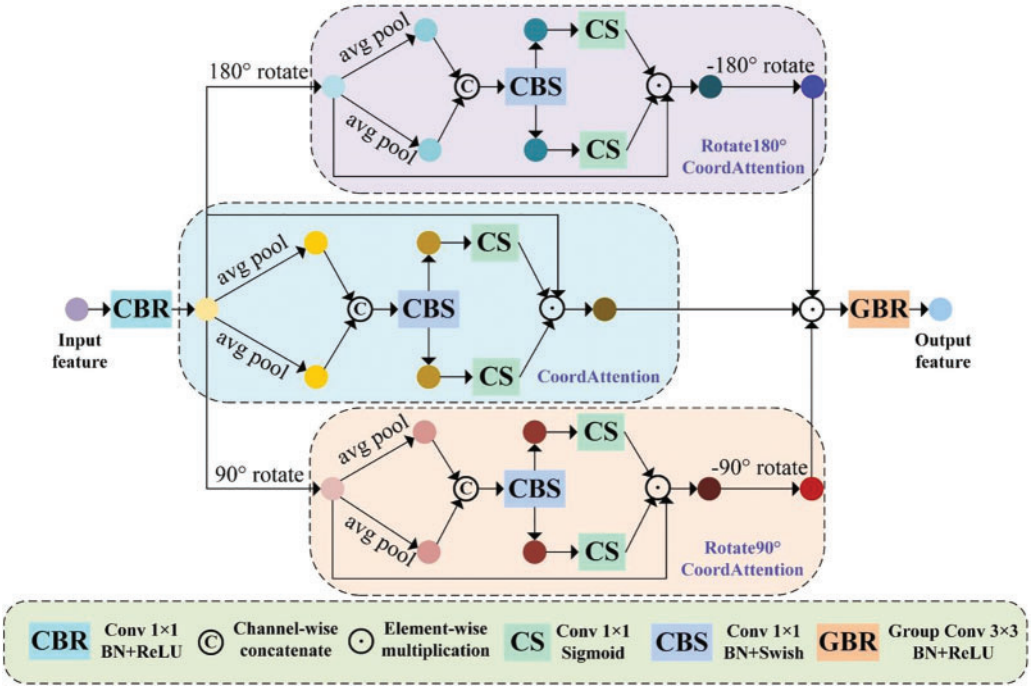

Figure 3: Multi-angle collaborative enhancement module

For the input features $f_{in} \in R^{C\times H\times W}$, $rotate\,90\,(f_{in},i)$ represents the result of rotating f_{in} clockwise by 90° for i times, as shown in Eq. (1).

$$f_{in}^{i} = rotate90\,(f_{in},i) \tag{1}$$

Eq. (2) represents the process of Coordinate Attention block. Eq. (3) represents rotating the feature maps back to the original angle.

$$f_{mid}^{i} = CoordAttention\left(f_{in}^{i}\right) \tag{2}$$

$$f_{out}^{i} = rotate90\left(f_{mid}^{i}, -i\right) \tag{3}$$

After aggregating the features from three angles, the final output of the MCE module is f_{out}, as shown in Eq. (4).

$$f_{out} = RConv_{3\times3}^{g=C_{in}}\left(f_{out}^{0} \odot f_{out}^{1} \odot f_{out}^{2}\right) \tag{4}$$

where $\odot$ denotes element-wise multiplication. $RConv_{3\times3}^{g=C_{in}}$ denotes a 3×3 group convolution grouped by the input channels with BN and ReLU activate function.

Under such design, the MCE module can efficiently enhance the representation of low-level features from three angles and aggregate detail differences to discover more rich detail cues of the glass surface.

3.2 LCFE Module

It is not easy to accurately localize the glass surface from the low-level information because many regions in the image have low-level features similar to the glass. Due to the transmission, refraction, and reflection of light by the glass material, there are pixel-level differences between the glass surface and its surroundings in the image, which makes the context of the glass and the background discontinuous. The combination of high-level semantic features on top of the low-level features can be used to better locate the glass surface region, so we propose the LCFE module in deep semantic understanding, consisting of two stages: extraction and selection. The LCFE module is shown in Fig. 4. The extraction stage draws on the structural features of the ASPP [22] module and the Densely connected Atrous Spatial Pyramid Pooling (DenseASPP) [36] module, where ASPP utilizes parallel dilated convolutions at different scales to increase the network receptive field, perceive the rich and wide range of contextual information and then locate different scales of glass surface features; the pixels on the glass surface are dense and tightly connected, DenseASPP can obtain denser multi-scale contextual information through serial dilated convolution, thus retaining more detailed information. Based on the advantages of the serial-connected and parallel-connected dilated convolution structures, in the feature extraction stage, we connect multiple depth-wise separable convolution units in a combination of parallel and serial, and capture the global semantic information through global average pooling. 3×3 DSC with a dilation rate is used for all dilated convolutions in the feature extraction stage, which reduces the number of parameters and computational consumption compared to the standard convolution [37], so this stage improves efficiency while retaining more detailed information and capturing rich contextual information at different scales to enhance the network's ability to perceive critical details. The different scale contexts extracted during the extraction stage are as follows:

$$f_1^E = Up\left(RConv_{1\times1}\left(AvgPool\left(RConv_{1\times1}\left(f_{in}\right)\right)\right)\right) \tag{5}$$

$$f_2^E = RDSC_{3\times3}^{d=1}\left(RConv_{1\times1}\left(f_{in}\right)\right) \tag{6}$$

$$f_3^E = RDSC_{3\times3}^{d=12}\left(RConv_{1\times1}\left(f_{in}\right)\copyright f_2^E\right) \tag{7}$$

$$f_4^E = RDSC_{3\times3}^{d=24}\left(RConv_{1\times1}\left(f_{in}\right)\copyright f_3^E\right) \tag{8}$$

$$f_5^E = RDSC_{3\times3}^{d=36}\left(RConv_{1\times1}\left(f_{in}\right)\copyright f_4^E\right) \tag{9}$$

where $f_{in} \in R^{C\times H\times W}$. $RConv_{1\times1}$ indicates 1×1 convolution with BN and ReLU. $RDSC_{3\times3}^{d=i}$ denotes 3×3 DSC with dilation rate i, BN and ReLU. © denotes concatenation in channel-wise.

We cascade the feature selection stage after the feature extraction stage to account for differences in the importance of context information at different scales. The Squeeze-and-Excitation Attention (SEAttention) [38] calculates feature weights channel-wise. However, it does not consider the inter-channel dependencies, where features from different channels may belong to the same context or different contexts. To obtain the weights of various scales of contextual information so that the network discovers differences in the importance of contexts and focuses on critical channels within the same context, a contextual weight branch parallel to the channel weight branch is added to the SEAttention for calculating the context-wise weights. In the selection stage, the output of the

extraction stage is subjected to adaptive average pooling and then fed into two parallel branches for the calculation of attention weights. Each branch comprises 1 × 1 convolution, ReLU, 1 × 1 convolution, and Sigmoid. The output dimensions of the two parallel branches are consistent with the context-wise and the channel-wise. $p_i \in [0, 1]$ and $q_i \in [0, 1]^C$ are context-wise weight and channel-wise weights, respectively, and finally, the process of attentional weighting is shown in Eq. (10).

$$f_i^S = p_i \cdot \left(f_i^E \otimes q_i\right) \tag{10}$$

where $\otimes$ denotes channel-wise multiplication. f_i^E is transformed into f_i^S after being weighted through attention.

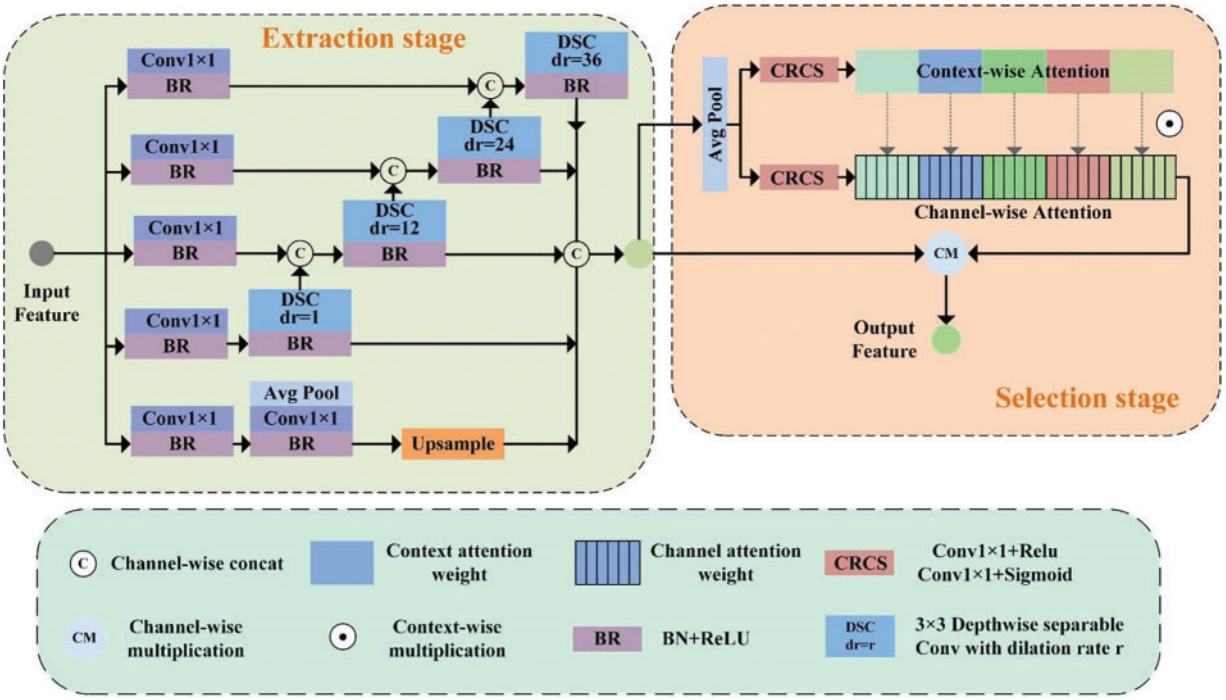

Figure 4: Large-scale contextual feature extraction module

3.3 Loss Function

To obtain finer glass surface segmentation results, we supervise the glass surface features and the boundary features at different stages, where the boundary tracking algorithm generates the boundary mask [39] before model training. Most computer vision tasks use cross-entropy loss as the loss function, but this function treats each pixel equally and ignores the image structure. Pixel position-aware loss [40] focuses on hard pixels and computes the structural information of the image, which is advantageous for glass surface segmentation tasks. With fewer pixels at the glass boundary, Dice loss can learn the category distribution and alleviate the problem of uneven category distribution [41]. Therefore, we use pixel position-aware loss as the loss function for glass surface segmentation and

Dice loss as the loss function for glass boundary extraction. The complete loss function is shown in Eq. (11).

$$Loss = \sum_{i=1}^{4} \frac{1}{2^{i-1}} PPA^{i} + Dice^{1} + \frac{1}{16} Dice^{6} \tag{11}$$

where PPA^{i} denotes the pixel position-aware loss between the glass surface segmentation result of stage i and the image label, and $Dice^{i}$ refers to the Dice loss between the glass boundary result extracted by stage i and the boundary label.

4 Experimental Results and Analysis

4.1 Experimental Environment

The configurations of the experimental environments are shown in Table 1.

Table 1: Experimental environment

Software and hardware	Configuration and version
CPU	Intel(R) Core(TM) i7-13700F CPU @ 5.20 GHz
GPU	Nvidia GeForce RTX 4090
Operating system	Ubuntu 18.04 LTS
Memory	32 G
Programming	Python 3.9
Framework	PyTorch 1.4, CUDA 9.1

During the training process, the backbone parameters are initialized with the pre-trained ResNeXt-101 of ImageNet, and the rest of the layers are randomly initialized; the SGD optimizer is used, with momentum set to 0.9 and weight decay set to 0.0005; the learning rate is adjusted by using the poly strategy, and the learning rate is initially set to 0.01; the batch size is 16, and the epoch is 200.

4.2 Datasets

Experiments are conducted on two open datasets, the home-scene-oriented glass segmentation dataset (HSO) [11] and the Trans10k-stuff dataset [6]. The images of HSO are taken from four home scene datasets and cropped to 1280 × 1024. The glass region in the dataset accounts for a concentrated proportion of images ranging from 0.2 to 0.6, with a uniform distribution of glass locations, where the training dataset contains 3070 images and the test dataset contains 1782 images. Trans10k-stuff is a transparent glass region dataset that includes most of the glass distribution scenes and is uncropped. The training set contains 2047 images, and the test set contains 1771 images.

To improve the model's generalization ability, the images are randomly rotated, mirrored, and blurred when the training data is loaded.

4.3 Evaluation Metrics

To fully validate the model, we use intersection over union ratio (IoU), mean absolute error (MAE), F-measure (F_β), and balance error rate (BER) as model evaluation metrics. IoU is the ratio of

the intersection over the union of the predicted mask to the true mask; MAE is the prime-level average error of the prediction mask to the real mask; F_β is the weighted ratio of precision and recall; BER is the ratio of errors in judging positive and negative classes in binary classification. The evaluation metrics formulas are as follows:

$$IoU = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(G(i,j) * P(i,j))}{\sum_{i=1}^{H}\sum_{j=1}^{W}(G(i,j) + P(i,j) - G(i,j) * P(i,j))} \tag{12}$$

$$MAE = \frac{1}{HW}\sum^{H}\sum^{W}|P(i,j) - G(i,j)| \tag{13}$$

$$F_\beta = \frac{(1+\beta^2)\, Precision \times Recall}{\beta^2 Precision + Recall} \tag{14}$$

$$BER = 1 - 0.5 \times \left(\frac{N_{tp}}{N_p} + \frac{N_{tn}}{N_n}\right) \tag{15}$$

where G and P denote the true mask and the predicted mask. H and W are the height and width of the true mask, respectively. β^2 is set to 0.3 [42], and N_{tp}, N_{tn}, N_p, and N_n denote the number of true, true-negative, glass, and non-glass pixels, respectively.

Mega Parameters (MParams) are the total number of parameters to be studied and optimized by the network and are used to measure the model's size. Giga Floating Point Operations Per Second (GFLOPs) indicate the model performs a quantity of floating-point operations per second during data processing and are utilized to evaluate the model's complexity. We apply MParams and GFLOPs to assess the model's efficiency.

4.4 Analysis of Experimental Results

To validate the effectiveness of our proposed method, we train the model on HSO and Trans10k-stuff datasets, respectively, and compare the model with binary semantic segmentation methods: PSPNet [43], DeepLabV3+ [22] and DenseASPP [36], mirror segmentation methods: MirrorNet [27] and HetNet [30], transparent object segmentation methods: TransLab [6] and Trans2Seg [7], and glass surface segmentation methods: GDNet [1], GSD [9] and PGSNet [11]. The comparison models are all trained to model convergence on the HSO train set and Trans10k-stuff train set, respectively, and tested on the HSO test set and Trans10k-stuff test set, respectively. The comparison results of different models on the HSO test set and Trans10k-stuff test set are shown in Table 2 and Table 3. The performance test results of the proposed method in this paper on the HSO test set and Trans10k-stuff test set are better than the other models. Compared to PGSNet, the MParams and GFLOPs of our model are reduced by 99.56 and 69, respectively, while performing better. In the HSO test set results, our model's F_β is improved by 1.1% over HetNet. In the Trans10k-stuff test set results, the IoU of our model is enhanced by 3.1% but the GFLOPs are reduced by 50.59 compared to TransLab. Experimental results show that our proposed method improves efficiency and performance.

Table 2: Comparison of results on the HSO test set

Model	IoU↑	F_β↑	MAE↓	BER↓	MParams↓	GFLOPs↓
PSPNet	0.776	0.814	0.095	0.106	67.95	95.44
DeepLabV3+	0.645	0.705	0.149	0.160	45.69	57.55
DenseASPP	0.759	0.805	0.096	0.113	46.07	55.85
MirrorNet	0.788	0.820	0.102	0.099	121.77	77.73
HetNet	0.796	0.832	0.089	0.093	49.92	27.77
TransLab	0.743	0.781	0.123	0.120	68.34	77.74
Trans2Seg	0.779	0.817	0.095	0.097	56.10	79.30
GDNet	0.787	0.817	0.097	0.093	201.72	231.54
GSD	0.789	0.818	0.103	0.098	83.72	92.60
PGSNet	0.801	0.836	0.089	0.091	142.27	96.15
Ours	**0.804**	**0.847**	**0.084**	**0.085**	**44.71**	**27.15**

Table 3: Comparison of results on the Trans10k-stuff test set

Model	IoU↑	F_β↑	MAE↓	BER↓	MParams↓	GFLOPs↓
PSPNet	0.879	0.907	0.045	0.055	67.95	95.44
DeepLabV3+	0.515	0.602	0.229	0.238	45.69	57.55
DenseASPP	0.863	0.894	0.051	0.061	46.07	55.85
MirrorNet	0.883	0.907	0.047	0.050	121.77	77.73
HetNet	0.893	0.915	0.046	0.048	49.92	27.77
TransLab	0.871	0.897	0.051	0.054	68.34	77.74
Trans2Seg	0.750	0.767	0.124	0.107	56.10	79.30
GDNet	0.886	0.907	0.046	0.047	201.72	231.54
GSD	0.897	0.917	0.042	0.045	83.72	92.60
PGSNet	0.898	0.917	0.042	0.044	142.27	96.15
Ours	**0.902**	**0.921**	**0.039**	**0.041**	**44.71**	**27.15**

4.5 Ablation Experiments

To further validate each module's effectiveness, we gradually add different modules to the network to validate each module. As shown in Table 4, the baseline is the network without adding any module. MCE, LCFE, and GB denote the addition of the MCE module, LCFE module, and SPPLEAN [32] module to the network, respectively. After adding the MCE module, the model's MParams and GFLOPs increase by only 0.02 and 0.06, but the IoU and F_β improve by 2.2% and 1.7%. Adding the LCEF module boosts the model's IoU and F_β by 1.5% and 1.1%, and adding the SPPLEAN module boosts the model's IoU and F_β by 1.4% and 1.6% but only increases the GFLOPs by 0.02. The ablation experiments' results show that the modules are efficient.

Table 4: Ablation experiment of modules

MCE	LCFE	GB	IoU↑	F_β↑	MParams↓	GFLOPs↓
			0.753	0.803	**43.41**	**26.86**
√			0.775	0.820	43.43	26.92
√	√		0.790	0.831	44.55	27.13
√	√	√	**0.804**	**0.847**	44.71	27.15

We adopt pixel position-aware loss [40] and Dice loss to compute the surface segmentation result loss and boundary extraction loss, respectively. To verify the difference between this hybrid loss function and the cross-entropy loss function or pixel position-aware loss alone, we use different loss functions to train and test our method on the HSO dataset, and the results are shown in Table 5. The model trained with our hybrid loss function shows improvements in IoU, MAE, and BER compared to the model trained with cross-entropy and pixel position-aware loss function.

Table 5: Comparison experiment of loss functions

Loss Function	IoU↑	F_β↑	MAE↓	BER↓
Cross entropy loss	0.801	0.846	0.085	0.087
Pixel position aware loss	0.803	**0.848**	0.089	0.089
Our hybrid loss	**0.804**	0.847	**0.084**	**0.085**

4.6 Effectiveness of Module Improvement

The MCE module inputs feature maps of different angles into three parallel coordinate attention blocks and fuses the feature differences through element-wise product and convolution. To verify the effectiveness of the MCE module compared to parallel coordinate attention blocks, we conduct the effectiveness experiments of the MCE module on the HSO dataset, as shown in Table 6. The MCE module outperforms multiple coordinate attention blocks, but the MParams and GFLOPs of the MCE module only increase by 0.01 and 0.02 over a coordinate attention block. The extraction stage of the LCFE module is improved based on the structure of the ASPP module and DenseASPP module, combining the advantages of the two connections and replacing the dilated convolutions with the DSCs with different dilation rates. The selection stage of the MCE module adds a context-wise weight calculation branch to SEAttention. As shown in Table 7, we validate the effectiveness of the LCFE module compared to the ASPP module combined with SEAttention and the DenseASPP module combined with SEAttention. Compared to the DenseASPP module combined with SEAttention, the LCFE module has a 4.67 reduction in MParams and a 0.3% improvement in IoU. LCFE module offers better performance with fewer MParams and GFLOPs. Experiments show that the MCE module and LCFE module improve performance and efficiency compared to the base module.

Table 6: Effectiveness of the MCE module

Module	IoU↑	F_β↑	MAE↓	BER↓	MParams↓	GFLOPs↓
CoordAttention	0.785	0.827	0.090	0.091	**44.70**	**27.13**
CoordAttention×2	0.790	0.831	0.089	0.090	44.71	27.14
CoordAttention×3	0.800	0.836	0.087	0.087	44.71	27.15
CoordAttention×4	0.798	0.842	0.086	0.088	44.71	27.15
MCE module	**0.804**	**0.847**	**0.084**	**0.085**	44.71	27.15

Table 7: Effectiveness of the LCFE module

Module	IoU↑	F_β↑	MAE↓	BER↓	MParams↓	GFLOPs↓
ASPP+SE	0.796	0.836	0.091	0.088	49.16	28.24
DenseASPP+SE	0.801	0.845	0.086	0.087	49.38	28.46
LCFE module	**0.804**	**0.847**	**0.084**	**0.085**	**44.71**	**27.15**

4.7 Comparison of Visualization Results

The visualization results of the proposed method in this paper and some comparison methods in the HSO test set and Trans10k-stuff test set are shown in Fig. 5. The red box indicates the deficiencies in the segmentation results. Visualization results show that our proposed method effectively improves the problems of pixel discontinuity and boundary roughness.

We take and label glass images from different scenes, which include transmission, reflection and refraction of light by the glass, to verify the generalization capacity of our proposed model trained on the HSO train set. Since glass surface segmentation is a binary classification task, the percentage of glass pixels in the captured image is about 1/2. The visualization results are displayed in Fig. 6, showing that our model has good generalization ability.

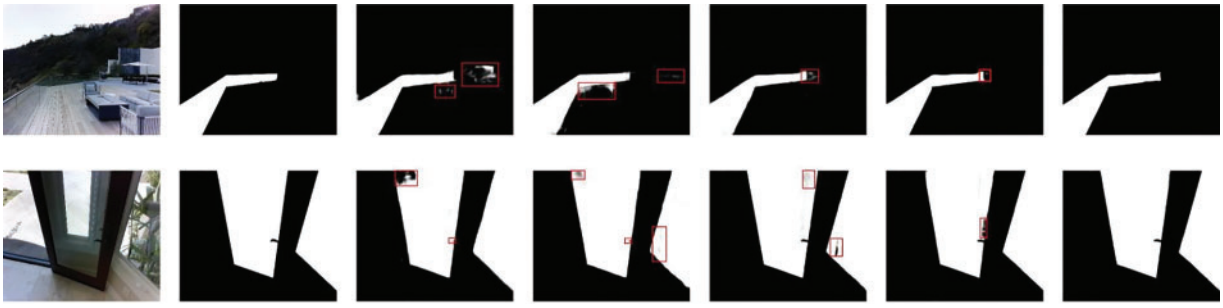

Figure 5: (Continued)

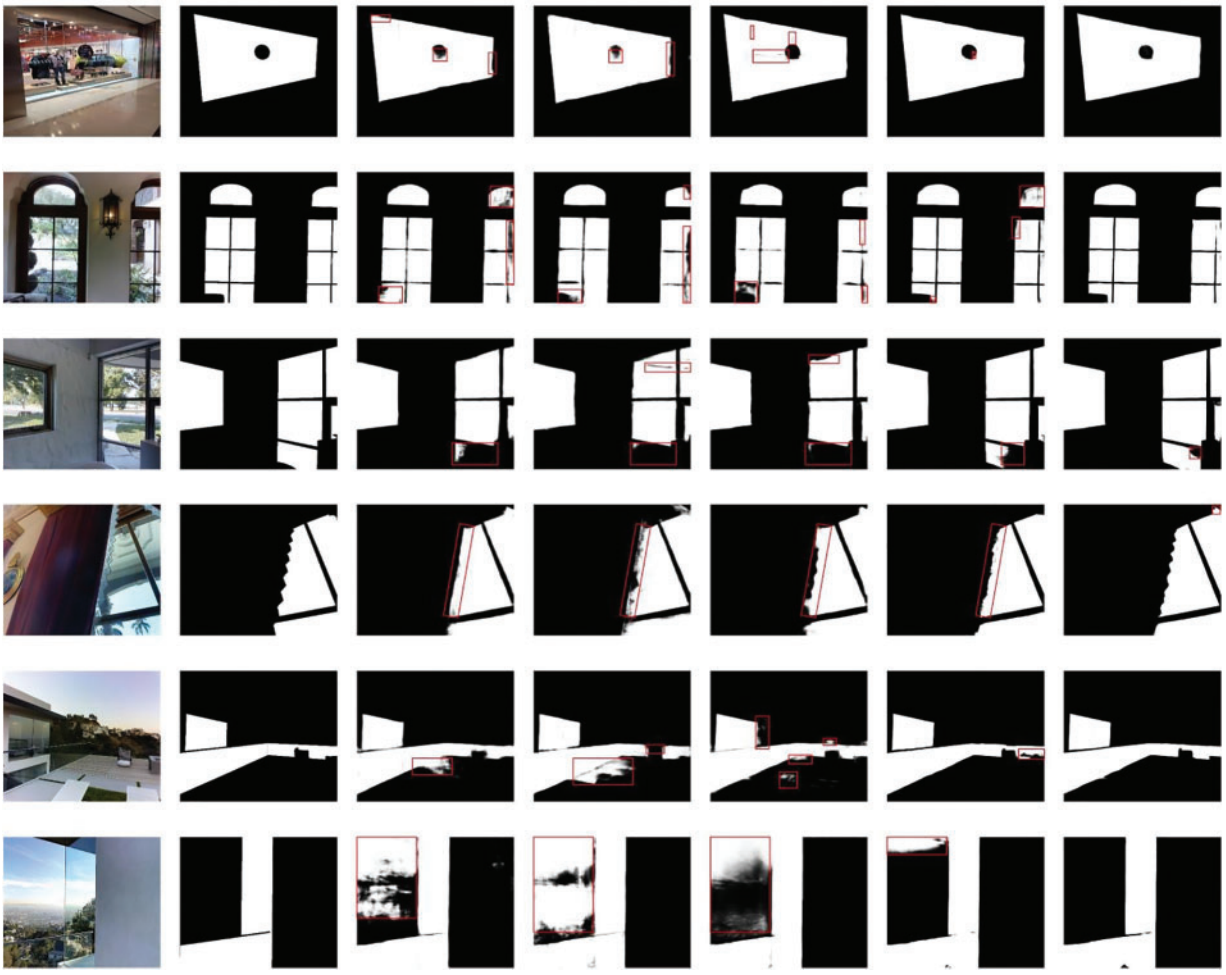

Figure 5: Different models' visualization results on the test set. From left to right 1-Glass image; 2-Ground-truth; 3-TransLab model; 4-DenseASPP model; 5-GDNet model; 6-HetNet model; 7-Our model

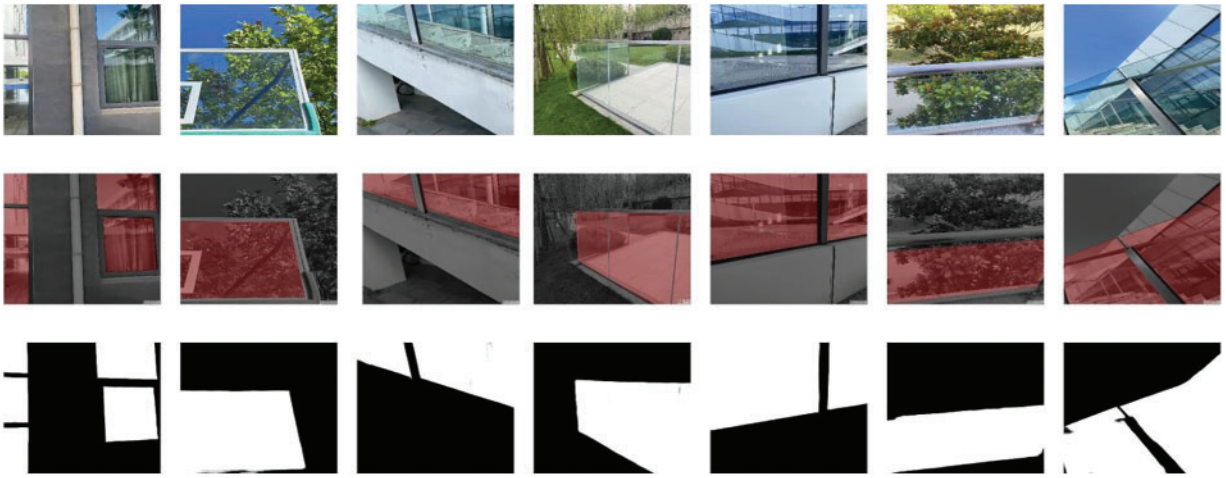

Figure 6: Our model's visualization results of images taken from different scenes. From above to below 1-Glass image; 2-Label; 3-Our model

5 Conclusions

To increase the efficiency of glass segmentation network and reduce the occurrence of boundary roughness and internal pixel discontinuities in the segmentation results, we propose an efficient glass segmentation network (EGSNet) based on multi-level heterogeneous architecture and boundary awareness. EGSNet employs different modules to deal with the information of various layers, thus equalizing the efficiency and performance of the network. For low-level understanding, the MCE module utilizes parallel coordinate attention blocks to enhance the detail representation from three angles, aggregate feature differences and explore richer detail cues. For semantic understanding, the LCFE module employs a series-parallel combined multi-scale context extraction structure, and attention weighting branches focusing on important contexts and channels, to adapt to the glass surface features in different scenarios. Under the level-by-level supervision and cross-level fusion of multi-level heterogeneous architecture, boundary features guide the orientation of glass contour; semantic understanding identifies the structural relationship between the glass surface and its surroundings; low-level understanding enriches the details of the glass surface; and features at all levels are fully fused to achieve accurate glass surface segmentation with the step-by-step refinement from the glass contour to the surface details. The results of experiments show that the efficiency of EGSNet is significantly higher than that of other glass segmentation models, while the performance is better than that of the comparison models. The visualization results of different scenes show that EGSNet has good generalization ability.

Nevertheless, the method put forward in this paper still possesses certain limitations. For example, in some images where the glass surface is occluded, the model may have detection errors; the segmented seams are not satisfactory enough when the arrangement between the glass pieces is denser. The future work will further explore the detailed clues of the glass surface and the difference in context information from the surroundings, design differentiated structures for the different angle branches of the MCE module, design a more effective context weight calculation branch in the selection stage of the LCFE module, and try to extract the glass boundaries at other stages. We will utilize the proposed model as teacher model to train a more efficient student model, which can be conveniently deployed

on devices with insufficient resources via knowledge distillation. To expand the dataset, we will shoot and label more glass images of different scenes, weather, and lighting conditions.

Acknowledgement: The authors would like to thank the editors and reviewers for their assistance and constructive comments.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm the following contributions to this paper: study conception and design: Tao Cui, Guojun Chen; data collection: Yongjie Hou, Huihui Li; analysis and interpretation of results: Tao Cui; draft manuscript preparation: Tao Cui, Yongjie Hou. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data presented in the study are available upon request from the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] H. Mei *et al.*, "Don't hit me glass detection in real-world scenes," in *2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 510–519. doi: 10.1109/CVPR42600.2020.00374.

[2] C. Xu *et al.*, "NeRF-Det: Learning geometry-aware volumetric representation for multi-view 3D object detection," in *2023 IEEE/CVF Int. Conf. Comput. Vis.(ICCV)*, Paris, France, 2023, pp. 23320–23330. doi: 10.1109/ICCV51070.2023.02131.

[3] A. -Q. Cao and R. de Charette, "MonoScene: Monocular 3D semantic scene completion," in *2022 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.(CVPR)*, New Orleans, LA, USA, 2022, pp. 3981–3991.

[4] G. Chen, K. Han, and K.-Y. K. Wong, "TOM-Net: Learning transparent object matting from a single image," in *2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 9233–9241.

[5] T. Sun, G. Zhang, W. Yang, J. -H. Xue, and G. Wang, "TROSD: A new rgb-d dataset for transparent and reflective object segmentation in practice," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 10, pp. 5721–5733, 2023. doi: 10.1109/TCSVT.2023.3254665.

[6] E. Xie *et al.*, "Segmenting transparent objects in the wild," in *Proc. of the European Conf. Comput. Vis. (ECCV)*, Glasgow, UK, 2020, pp. 696–711.

[7] E. Xie *et al.*, "Segmenting transparent objects in the wild with transformer," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Montreal, QC, Canada, 2021, pp. 1194–1200.

[8] H. Mei, X. Yang, L. Yu, Q. Zhang, X. Wei and R. W. H. Lau, "Large-field contextual feature learning for glass detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3329–3346, 2023. doi: 10.1109/TPAMI.2022.3181973.

[9] J. Lin, Z. He, and R. W. H. Lau, "Rich context aggregation with reflection prior for glass surface detection," in *2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, 2021, pp. 13410–13419.

[10] F. Liu, Y. Liu, J. Lin, K. Xu, and R. W. H. Lau, "Multi-view dynamic reflection prior for video glass surface detection," in *Proc. AAAI Conf. Artif. Intell.*, Washington, DC, USA, 2023, pp. 3594–3602.

[11] L. Yu *et al.*, "Progressive glass segmentation," *IEEE Trans. Image Process.*, vol. 31, pp. 2920–2933, 2022. doi: 10.1109/TIP.2022.3162709.

[12] H. Mei *et al.*, "Glass segmentation using intensity and spectral polarization cues," in *2022 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, 2022, pp. 12612–12621.
[13] D. Huo, J. Wang, Y. Qian, and Y. -H. Yang, "Glass segmentation with RGB-thermal image pairs," *IEEE Trans. Image Process.*, vol. 32, pp. 1911–1926, 2023. doi: 10.1109/TIP.2023.3256762.
[14] H. He *et al.*, "Enhanced boundary learning for glass-like object segmentation," in *2021 IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, 2021, pp. 15839–15848.
[15] Y. Lin, Y. Yeung, and R. W. H. Lau, "Exploiting semantic relations for glass surface detection," in *Proc. Int. Conf. Neural Inform. Process. Syst. (NIPS)*, New Orleans, LA, USA, 2022, pp. 22490–22504.
[16] J. Zhang, G. Yang, and C. Liu, "DCNet: Glass-like object detection via detail-guided and cross-level Fusion," in *Proc. Int. Conf. Intell. Comput.*, Zhengzhou, China, 2023, pp. 461–472.
[17] F. Qi *et al.*, "Glass makes blurs: Learning the visual blurriness for glass surface detection," *IEEE Trans. Ind. Inform.*, vol. 20, no. 4, pp. 6631–6641, 2024. doi: 10.1109/TII.2024.3352232.
[18] X. Tan, F. Qi, N. Wang, Z. Zhang, Y. Xie and L. Ma, "Glass surface detection method based on visual distortion," *J. Comput.-Aided Design Comput. Graph.*, 2023. doi:10.3724/SP.J.1089.2023-00342.
[19] C. Zheng, P. Li, X. Zhang, X. Lu, and M. Wei, "Don't worry about mistakes! glass segmentation network via mistake correction," 2023, *arXiv:2304.10825*.
[20] X. Qin, J. Liu, Q. Wang, S. Zhang, F. Zhu and Z. Yi, "Fourier boundary features network with wider catchers for glass segmentation," 2024, *arXiv:2405.09459*.
[21] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *2017 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 5168–5177. doi: 10.1109/CVPR.2017.549.
[22] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. European Conf. Comput. Vis. (ECCV)*, Munich, Germany, 2018, pp. 801–818.
[23] Y. Cao, Z. Zhang, and E. Xie, "Fakemix augmentation improves transparent object detection," 2021, *arXiv:2103.13279*.
[24] T. Senthivel and N. -S. Vu, "Subgroups for detection transformer," in *2024 IEEE Int. Conf. Image Process. (ICIP)*, 2024, pp. 2194–2200. doi: 10.1109/ICIP51287.2024.10648285.
[25] C. Wu, W. Liu, and X. Ma, "Infrared and visible image fusion based on Res2Net-transformer automatic encoding and decoding," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 1441–1461, Apr. 2024. doi: 10.32604/cmc.2024.048136.
[26] S. Zheng *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, 2021, pp. 6877–6886. doi: 10.48550/arXiv.2012.15840.
[27] X. Yang, H. Mei, K. Xu, X. Wei, B. Yin and R. Lau, "Where is my mirror?" in *2019 IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, Republic of Korea, 2019, pp. 8808–8817. doi: 10.1109/ICCV.2019.00890.
[28] J. Lin, G. Wang, and R. W. H. Lau, "Progressive mirror detection," in *2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 3694–3702.
[29] T. Huang, B. Dong, J. Lin, X. Liu, R. Lau and W. Zuo, "Symmetry-aware transformer-based mirror detection," in *Proc. AAAI Conf. Artif. Intell.*, Washington, DC, USA, 2023, pp. 935–943.
[30] R. He, J. Lin, and R. W. H. Lau, "Efficient mirror detection via multi-level heterogeneous learning," in *Proc. AAAI Conf. Artif. Intell.*, Washington, DC, USA, 2023, pp. 790–798.
[31] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conf. Comput. Visi. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 5987–5995. doi: 10.1109/CVPR.2017.634.
[32] C. -Y. Wang, I. H. Yeh, and H. -Y. M. Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," 2024, *arXiv:2402.13616*.
[33] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 1800–1807.

[34] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, 2021, pp. 13708–13717. doi: 10.1109/CVPR46437.2021.01350.

[35] S. Seo, J. Oh, E. Cho, H. Park, G. Kim and J. Kim, "TP-MobNet: A two-pass mobile network for low-complexity classification of acoustic scene," *Comput. Mater. Contin.*, vol. 73, no. 2, pp. 3291–3303, Jun. 2022. doi: 10.32604/cmc.2022.026259.

[36] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 3684–3692. doi: 10.1109/CVPR.2018.00388.

[37] Q. Tong, Z. Zhu, M. Zhang, K. Cao, and H. Xing, "CrossFormer embedding deeplabv3+ for remote sensing images semantic segmentation," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 1353–1375, Apr. 2024. doi: 10.32604/cmc.2024.049187.

[38] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 7132–7141. doi: 10.1109/CVPR.2018.00745.

[39] Y. Qin, N. Zhao, J. Yang, S. Pan, B. Sheng and R. W. H. Lau, "UrbanEvolver: Function-aware urban layout regeneration," *Int. J. Comput. Vis. (IJCV)*, vol. 132, pp. 3408–3427, 2024. doi: 10.1007/s11263-024-02030-w.

[40] J. Wei, S. Wang, and Q. Huang, "F^3Net: Fusion, feedback and focus for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, Washington, DC, USA, 2020, pp. 12321–12328.

[41] C. He *et al.*, "Camouflaged object detection with feature decomposition and edge reconstruction," in *2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Vancouver, BC, Canada, 2023, pp. 22046–22055. doi: 10.1109/CVPR52729.2023.02111.

[42] J. Li, Z. Wang, Z. Pan, Q. Liu, and D. Guo, "Looking at boundary: Siamese densely cooperative fusion for salient object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 7, pp. 3580–3593, 2023. doi: 10.1109/TNNLS.2021.3113657.

[43] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *2017 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 6230–6239. doi: 10.1109/CVPR.2017.660.