


Article

TGSNet: Multi-Field Feature Fusion for Glass Region Segmentation Using Transformers

Xiaohang Hu ¹, Rui Gao ¹, Seungjun Yang ² and Kyungeun Cho ^{1,*}¹ Department of Multimedia Engineering, Dongguk University-Seoul, 30, Pildongro-1-gil, Jung-gu, Seoul 04620, Republic of Korea² Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Republic of Korea

* Correspondence: cke@dongguk.edu

Abstract: Glass is a common object in living environments, but detecting it can be difficult because of the reflection and refraction of various colors of light in different environments; even humans are sometimes unable to detect glass. Currently, many methods are used to detect glass, but most rely on other sensors, which are costly and have difficulty collecting data. This study aims to solve the problem of detecting glass regions in a single RGB image by concatenating contextual features from multiple receptive fields and proposing a new enhanced feature fusion algorithm. To do this, we first construct a contextual attention module to extract backbone features through a self-attention approach. We then propose a ViT-based deep semantic segmentation architecture called MFT, which associates multilevel receptive field features and retains the feature information captured by each level of features. It is shown experimentally that our proposed method performs better on existing glass detection datasets than several state-of-the-art glass detection and transparent object detection methods, which fully demonstrates the better performance of our TGSNet.

Keywords: glass detection; transformer; feature fusion algorithm; image classification

MSC: 68T45; 68T07; 68U10



Citation: Hu, X.; Gao, R.; Yang, S.; Cho, K. TGSNet: Multi-Field Feature Fusion for Glass Region Segmentation Using Transformers. *Mathematics* **2023**, *11*, 843. <https://doi.org/10.3390/math11040843>

Academic Editor: Vassilios Solachidis

Received: 3 January 2023

Revised: 30 January 2023

Accepted: 3 February 2023

Published: 7 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Glass is widely used; it is made into windows, door frames, appliances, decorations, lamps, furniture, and other items, and it is an indispensable resource. However, glass can pose problems, such as when robots fail to recognize glass, leading to collision, and 3D point cloud reconstruction fields generate considerable noise owing to the effect of glass reflections. Therefore, the detection of glass is essential, and it is a new and difficult computer vision challenge. There are many types of glass, such as common flat glass, nondual-transparent matte glass, and colorful fancy window glass. These types of glass can be affected by the surrounding environment and lighting conditions to show different characteristics, and their patterns, scattering, reflection, and color make them difficult to detect. At the same time, the presence of glass has a significant impact on many other areas of research, such as point cloud denoising tasks [1], depth estimation algorithms [2,3], salient object detection methods [4–6], and semantic segmentation methods [7–14]. Owing to the nature of the glass itself, it can easily be confused with the scene, resulting in low performance on these tasks in scenes with a large amount of glass.

To solve the impact caused by glass, the first task is to collect a large amount of image data containing glass objects and glass-like objects to construct datasets; authors of [15–17] accomplished the task of constructing glass datasets. Based on these datasets, great progress has been made in the fields of semantic segmentation and salient object detection for detecting glass in 2D images [18–22], and our TGSNet is implemented based on these datasets. We designed a semantic segmentation method to implement glass

detection. Many glass region detection methods that depend on extra data have been used to detect glass in images, such as using heat maps (RGB-T) [23], depth maps (RGB-D) [24], and polarization maps (RGB-P) [25]. Although these studies achieved good results, their data requirements are rigorous, and their methods are not applicable if only RGB images are used. In other methods that use only RGB images, authors of [15] designed a structure for extracting different layer contexts from backbone features to detect different sizes of glass in a scene, and authors of [26] devised an enhanced boundary learning method to detect glass-like objects. However, the experimental results of these methods can be improved. To solve these problems, we believe that using a vision transformer (ViT) is a good solution, where ViT [12] provides a new idea for migrating natural language processing (NLP) tasks to computer vision tasks, which can achieve good results with fewer training resources. However, the limitations of ViT are that it cannot handle different input sizes, and its position encoding has a fixed-length limitation.

To solve the problem of detecting glass objects from a single RGB image and the limitations of ViT mentioned above, we propose TGSNet, which is a transformer-based glass region segmentation network. First, we design a contextual attention module (CAM) that uses self-attention to concatenate the multilayer results obtained from the feature backbone. Then, we design a network called a multifield transformer (MFT) based on the ViT model, which can provide different receptive field outputs. Because the features of glass objects are usually difficult to capture, we associate multiple receptive fields to infer the glass regions and retain the feature information obtained at each level of the receptive field. Finally, to fully utilize the feature information at each level of receptive field, a cross-modal contextual feature fusion (CCFF) module consisting of multiple transformer heads is built into TGSNet. First, the features are separated cross-modally into segmentation features and boundary features and analyzed separately. Then, a multiheaded self-attention-based transformer and dilation convolution are used for multi-size perceptual field feature fusion. Dilation convolution is a deep structural extension of our atrous spatial pyramid pooling module (ASPP) in DeepLabv3 [18], whereby we designed the unique cross-modal ASPP (C-ASPP). It allows the analysis of boundary features and segmentation features separately, inspired by AdaptiveASPP [19].

Effectively, when we used ResNeXt101 as the backbone network, the method presented in this paper showed significant performance improvement, similar to other existing methods. Our method can visually detect glass objects with more detailed boundaries and produce more accurate segmentation results on a glass detection dataset (GDD) [15] than can existing methods.

Concisely, the contribution of our proposed method is as follows:

- We first construct a CAM to extract backbone features through a self-attention approach. We then propose a ViT-based deep semantic segmentation architecture, called MFT, which associates multilevel receptive field features and retains the feature information captured by each level of features.
- A CCFF module is designed. It can extract boundary and segmentation features from multiscale and cross-modal features and associate contextual field-of-view fusion features.
- A model named TGSNet is constructed, which outperforms existing glass detection methods in terms of both performance and visual performance.

2. Related Work

The essence of the glass detection task is to label the glass regions in an image, usually using the glass and background as labels for classification; therefore, the core problem is the same as semantic segmentation. This can be regarded as a semantic segmentation subtask.

Semantic segmentation: Semantic segmentation associates each pixel in an image with its corresponding class (label) to classify the image content. Some early approaches [27,28] used fully convolutional networks and proposed the concepts of feature map fusion and stitching. Badrinarayanan et al. [29] optimized the fully connected layer using an encoder–

decoder structure and proposed the use of a maximum pooling index for up-sampling. Chen et al. [18,30,31] proposed a null pyramid pooling method in combination with the Zhao et al. pyramid pooling module [10], which uses null convolution to expand the receptive fields of view and refine the boundaries. He et al. [8] extended Fast R-CNN [32] using binary segmentation and proposed the use of bilinear interpolation to upsample the features for accuracy.

Transformer for semantic segmentation: In recent years, transformers have commonly been used in the computer vision field. Wang et al. [11,33] designed a pure transformer backbone and developed a pyramidal structure of attention layers that could save computational resources and detect multiscale features. Guo et al. [34] designed a convolutional attention module that saves a large amount of computational resources compared with self-attention while guaranteeing good performance. Xie et al. [16] provided a large transparent object dataset containing glass types and designed an encoder–decoder network that could provide a global receptive field and classify glass region based on the ViT network [12]. Zhang et al. [35] proposed a deeper encoder–decoder network and designed a small transformer head to prevent overfitting.

Glass detection: Recently, Mei et al. [15] contributed to a glass detection dataset, GDD, and proposed a method to extract different layer features from the backbone network and fuse deep and shallow features to detect glass. Cao et al. [19] proposed a method to enhance the ability to distinguish boundaries that could extract features across modalities and multiple scales. Hao et al. [26] proposed a boundary-aware module that could model the boundaries of global shapes. Some methods use multidimensional data; for example, Mei et al. [20] used polarization information to detect glass. Huo et al. [23] used thermograms to fuse RGB images and thermal modal features to detect glass. Lin et al. [24] constructed an RGB-D glass dataset and used depth information to analyze glass features.

After reviewing existing methods in related fields, we found that there was room for improvement. Some approaches concatenate contextual features but ignore the differences between different levels of features. There are also approaches that use a combination of boundary and segmentation features, but they ignore the holistic nature of the target features. By combining the advantages of the above approaches and summarizing the drawbacks of the current methods, we propose TGSNet. Compared with previous works, our backbone uses a multilayer transformer based on convolutional attention and designs a novel feature fusion module consisting of transformer heads.

3. Proposed Method

Our approach combines the strengths of and addresses the weaknesses of existing state-of-the-art techniques. We first design the CAM module, which extracts and connects contextual information from the backbone network with features of different sizes using self-attention. We use the convolutional transformer structure in MFT to obtain richer feature semantics from different receptive fields. Therefore, deeper structures must be used to achieve a greater number of receptive field scales. However, when we use self-attention with a multilayer transformer structure, the number of parameters increases as the structure deepens. To solve this problem, we chose to use multiscale convolutional attention, and multiscale convolutional attention structures ensure the use of a smaller number of parameters while ensuring as much accuracy as possible [18]. The CCFF module aims to transform feature information into cross-modality boundary and segmentation features and enhance both features. The background feature information is then gradually cascaded from a large receptive field to a small receptive field.

As shown in Figure 1, our network consists of four parts: a feature backbone, CAM, MFT backbone, and CCFF module. The base feature backbone network we use is ResNeXt-101, with an output of four layers of size, $(\frac{H}{4}, \frac{W}{4}, 256)$, $(\frac{H}{8}, \frac{W}{8}, 512)$, $(\frac{H}{16}, \frac{W}{16}, 1024)$, and $(\frac{H}{32}, \frac{W}{32}, 2048)$ features. CAM contains four sets of attention-transforming header blocks for obtaining the output of the feature backbone network, as described in Section 3.1. The MFT contains six encoders to obtain receptive field features of different sizes, as described in

Section 3.2. The CCFF multiscale attention to the adaptive fusion of different receptive field features is described in Section 3.3.

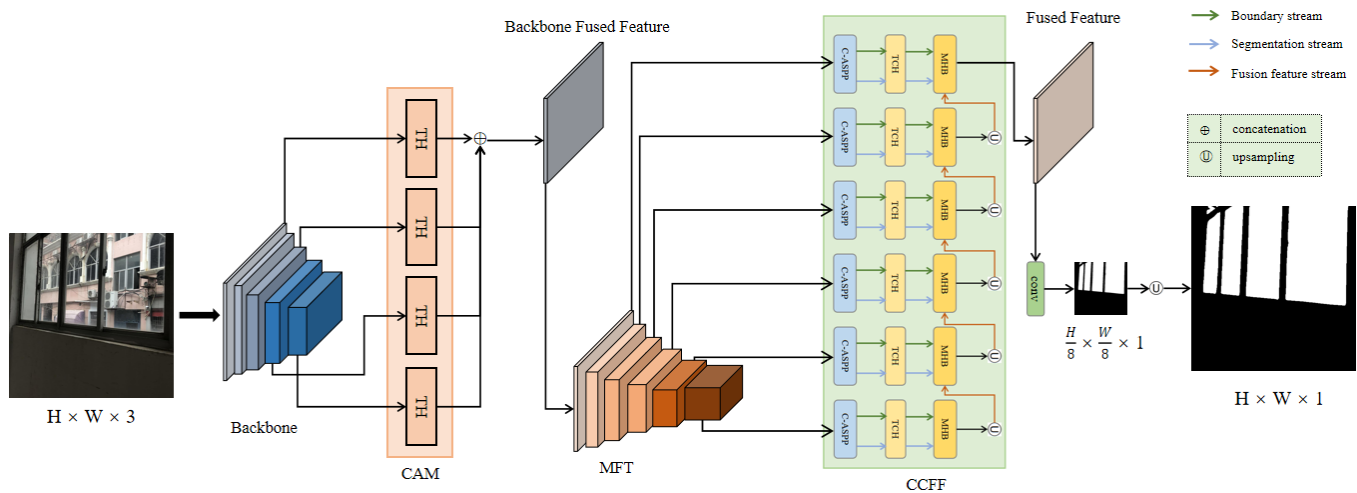


Figure 1. Overall framework structure of TGSNet.

3.1. CAM

Extracting the features of a backbone network is usually done with the convolution method to obtain feature information at different scales, but the convolution method limited receptive field prevents them from aggregating global context [36]. Therefore, to obtain more accurate features, we designed a CAM that, as the name suggests, uses four groups of light transformer header blocks in parallel, as shown in Figure 2, to extract features from the backbone network output from the four layers of contextual information to obtain preliminary backbone fusion features. The purpose of doing this is to weight the target region, so the self-attention can focus on important information and ignore useless information [37,38]. The role of the parallel structure in this study is to reduce the differences that exist between the feature information that is output by different layers to suppress useless information and to enable global and local linkage [12], as specified in Section 4.5 (A). In each transformer header block, a multiheaded self-attention layer and feed-forward network (FFN) layer are included, which are used for the multiheaded self-attention layer to add weights to the features, and FFN converts all the shapes of the features to $(\frac{H}{4}, \frac{H}{4}, 64)$. The output shape of the first layer is $(\frac{H}{4}, \frac{H}{4}, 64)$. The output shape of the second layer is $(\frac{H}{8}, \frac{H}{8}, 64)$. The third layer output shape is $(\frac{H}{16}, \frac{H}{16}, 64)$, and the fourth layer output shape is $(\frac{H}{32}, \frac{H}{32}, 64)$. These shapes are all transformed, upsampled to $(\frac{H}{4}, \frac{H}{4}, 64)$, and fused with the features of the previous layer. The attention weighting process of each layer can be expressed as:

$$head_i = \text{Attention}(q_i, K, V) \quad (1)$$

$$\text{Attention}(q_i, K, V) = \text{softmax}\left(\frac{q_i K^T}{\sqrt{d_k}}\right) V \quad (2)$$

$$\text{MultiHeadAttention}(Q, K, V) = \text{Concat}(head_1, head_2, \dots, head_n) W \quad (3)$$

where $q_i \in \mathbb{R}^{d_q}$ is the query for each layer, $K \in \mathbb{R}^{d_k}$ is the key, $V \in \mathbb{R}^{d_v}$ is the value, d_k is the scaling factor, and W is the learnable parameter.

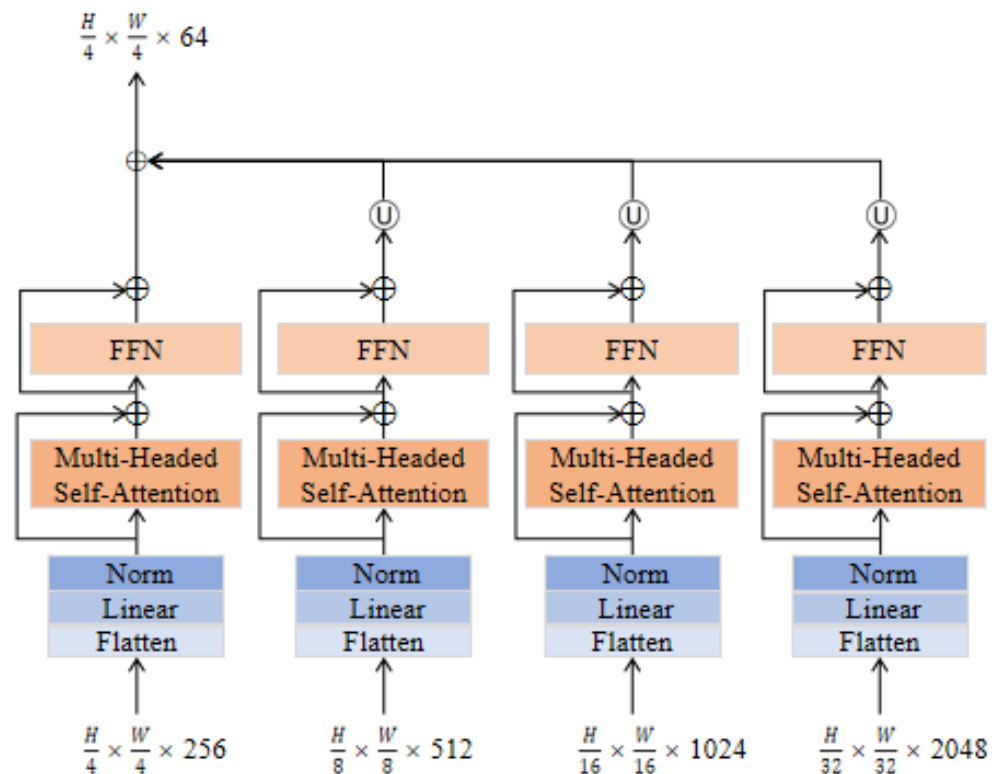


Figure 2. Contextual attention module structure.

After multi-head self-attention layer weighting and FFN transformation to a specified number of channels, we have 64 channels. The feature map obtained from each layer is upsampled to the size of the first layer using a bilinear interpolation method. The feature maps of all layers are fused, and the operation process can be expressed as follows:

$$F_{CAM} = \text{contact}(F_{T1}, F_{T2}, F_{T3}, F_{T4}) \quad (4)$$

F_{CAM} represents the attention-weighted backbone features obtained by fusion, F_{Ti} denotes the weighted features of each layer, and “contact” denotes the feature fusion step.

3.2. MFT

After the backbone features are fused, the features have redundant information because of the different features in the different layers [39,40]. Therefore, we designed a multilevel field-converter backbone with the structure shown in Figure 3. The main purpose was to fuse the backbone features obtained from CAM one by one with six different receptive fields, $(\frac{H}{4}, \frac{W}{4}, 64)$, $(\frac{H}{8}, \frac{W}{8}, 128)$, $(\frac{H}{16}, \frac{W}{16}, 192)$, $(\frac{H}{16}, \frac{W}{16}, 256)$, $(\frac{H}{32}, \frac{W}{32}, 320)$, and $(\frac{H}{32}, \frac{W}{32}, 512)$, for feature analysis, where H and W refer to the length of the input features in the longitudinal and lateral directions, respectively; the smaller the input size, the larger the receptive field. We obtain more details about the local features from the small receptive field and then correlate them downward to realize the local-to-global concatenation. The entire feature analysis process is performed sequentially from the small to the large receptive fields. The input of the first layer is the fused features of the CAM, and the input to layers 2–6 is the output of the previous layer. The output of each layer is stored and input to the next module, while the features of each layer are retained.

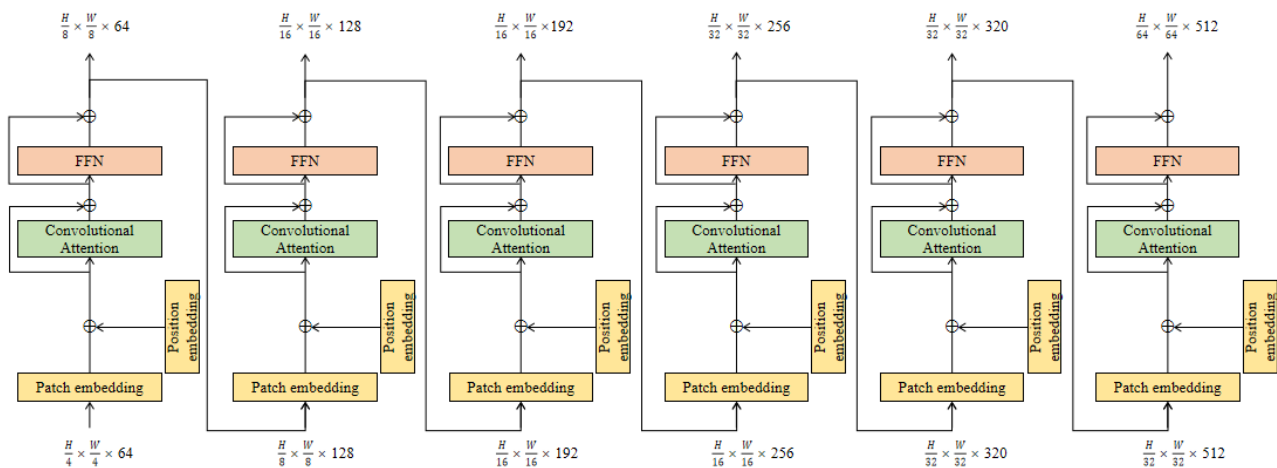


Figure 3. Multilevel field transformer structure.

We follow the overall structure of the traditional visual transformer, which contains patch embedding, position embedding, an attention block, and a feedforward network. However, unlike ViT [12], as shown in Figure 3, a convolutional attention block [34] instead of a multiheaded self-attention block is used in this study to construct a new convolutional attention transformer, whose internal details are shown in Figure 4.

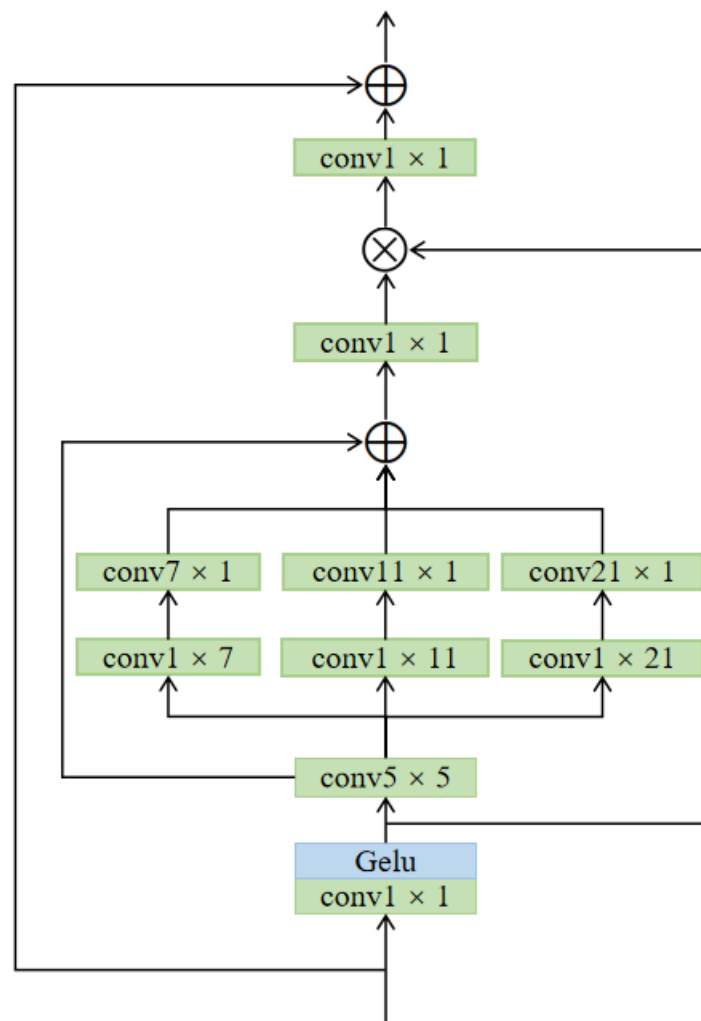


Figure 4. Convolutional attention module.

This convolutional attention module contains several main parts. One is the cross-channel linear combination of input feature information that uses 1×1 convolution, that is, linear transformation along different channels, to integrate the information between different channels. Second, for multiscale contextual feature integration, we choose to use a 5×5 convolution to guarantee an effective receptive field [41]. Although using two 3×3 convolutional stacks reduces the number of parameters, the proper use of large convolutional kernels can strengthen localization and classification ability [42]. Subsequently, convolutional attention is divided into $(1 \times 7, 7 \times 1)$, $(1 \times 11, 11 \times 1)$, and $(1 \times 21, 21 \times 1)$ convolutional sizes. Dividing a two-dimensional convolution into a series of asymmetric convolutions can effectively reduce the total number of parameters to a great extent, while the banded convolution facilitates the analysis of narrowly shaped targets [43], which also ensures accuracy. Therefore, we use three asymmetric convolutional kernels of different sizes instead of multiheaded self-attention, and the process of multiscale attentional analysis from can be expressed as:

$$atten = Conv_{5 \times 5}(F) \quad (5)$$

$$atten_1 = Conv_{7 \times 1}(Conv_{1 \times 7}(atten)) \quad (6)$$

$$atten_2 = Conv_{11 \times 1}(Conv_{1 \times 11}(atten)) \quad (7)$$

$$atten_3 = Conv_{21 \times 1}(Conv_{1 \times 21}(atten)) \quad (8)$$

where F is the feature obtained by channel integration, “ $atten_1$ ” is the attention weight matrix obtained by the convolution of size $(1 \times 7, 7 \times 1)$, “ $atten_2$ ” is the attention weight matrix obtained by the convolution of size $(1 \times 11, 11 \times 1)$, “ $atten_3$ ” is the attention weight matrix obtained by the convolution of size $(1 \times 21, 21 \times 1)$, and “ $Conv_{n \times n}$ ” is the convolution size.

The three obtained weight outputs are fused with the attention weights of the 2D convolutional output, and a convolutional attention weight matrix is then output. The feature “ F ” will then be multiplied with the attention weighting matrix attention, as in the self-attention mechanism, to obtain the attention-weighted features F_{atten} .

$$Attention = atten + atten_1 + atten_2 + atten_3 \quad (9)$$

$$F_{atten} = F \otimes Attention \quad (10)$$

This operation considerably reduces the number of parameters without an excessive loss of accuracy. Subsequently, the module integrates the channel information using a 1×1 convolution and correlates it with the original features. Finally, the convolutional attention module outputs a feature with the convolutional attention weights.

3.3. CCFF

The CCFF module, whose structure is shown in Figure 5, contains six groups of feature fusion modules composed of three parts: C-ASPP, transformer conversion head (TCH), and mixing head block (MHB). The purpose of the C-ASPP is to transform the input feature information cross-modally into boundary and segmentation features. TCH reuses self-attention weighting for the obtained boundary and segmentation features. MHB fuses the features of the previous layer and reuses the self-attention weighting for the fusion results. The input of each group comes from the output of MFT, and its top-to-bottom input shapes are $(\frac{H}{8}, \frac{W}{8}, 64)$, $(\frac{H}{16}, \frac{W}{16}, 128)$, $(\frac{H}{16}, \frac{W}{16}, 192)$, $(\frac{H}{32}, \frac{W}{32}, 256)$, $(\frac{H}{32}, \frac{W}{32}, 320)$, and $(\frac{H}{64}, \frac{W}{64}, 512)$.

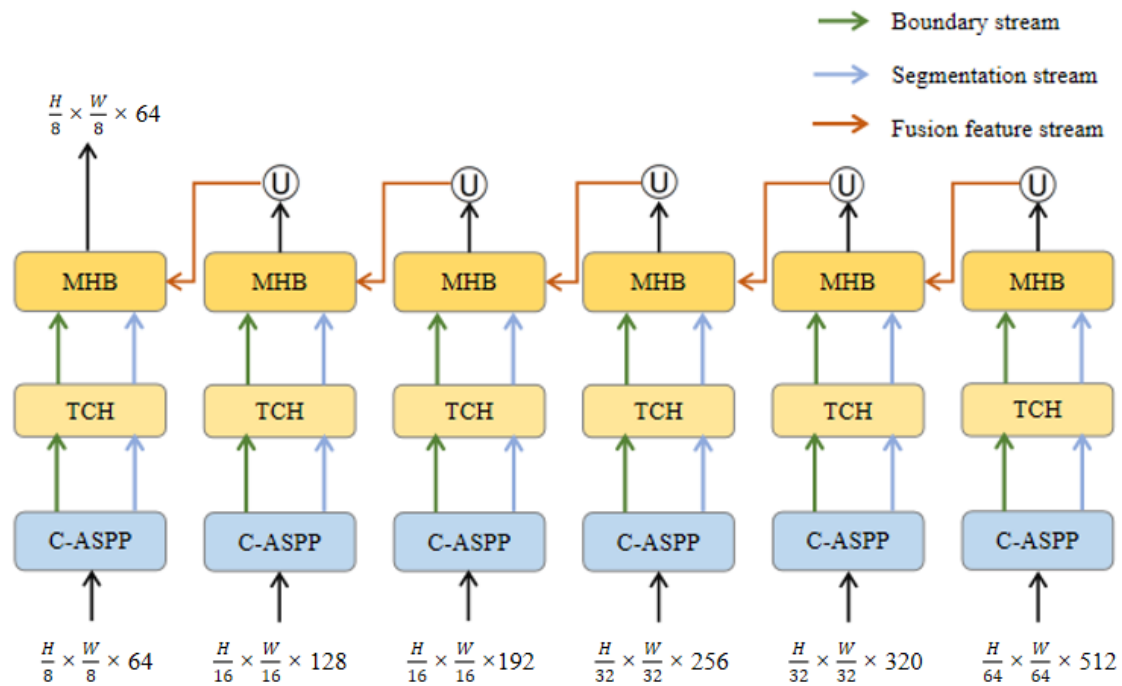


Figure 5. Cross-modal contextual feature fusion module.

To avoid losing accuracy by downsampling and to generate redundant computations by repeatedly convolving the same region, we invoke a dilation convolution method called ASPP [31]. Our C-ASPP also benefits from the structure of the cross-modal analysis boundary and segmentation features in AdaptiveASPP [19], which uses dilation convolution with five dilation rates. Owing to the multilevel structure of our MFT, we further extend ASPP [31] to ensure that each receptive field can obtain sufficient features. The C-ASPP structure shown in Figure 6 was also designed with a dilation convolution structure having six dilation rates of 6, 12, 18, 24, 30, and 36 steps. Dilation rates can serve to increase the receptive field without downsampling to analyze a larger range of feature information. First, features are input to C-ASPP to obtain boundary and segmentation features across modalities to obtain more information about the scaled receptive field. We use the ASPP [31] to obtain multiscale features with different dilation rates and to preserve the overall data features and prevent overfitting using adaptive, two-dimensional averaging pooling:

$$F_n = \text{SeparableConv}2d_n(F_{atten}) \quad (11)$$

$$f_n = \text{AdaptiveAvgPool}2d(F_n) \quad (12)$$

where F_n denotes the feature acquired at dilation rate n . $\text{SeparableConv}2d_n$ is the depth-separable convolution method used to implement void convolution, and it corresponds to the dilation rate. $\text{AdaptiveAvgPool}2d$ is an adaptive two-dimensional averaging pooling method, F_{atten} denotes the feature output from the corresponding layer of MFT, and f_n denotes the feature after average pooling at the n dilation rate.

However, the features of the detection boundary differ from those of the detection target region. Therefore, we use a branch of boundary feature extraction to extract the boundary and segmentation features separately at multiple scales. This section references the AdaptiveASPP approach [19], where the boundary modal branch and the segmentation modal branch are feature-enhanced using the original features separately, and a residual layer is added to prevent network degradation. The process of feature-enhanced can be expressed separately as:

$$S = R(\text{concat}(\sum_{n=1}^6 \phi(f_n^s) + F_n)) \quad (13)$$

$$B = R(\text{concat}(\sum_{n=1}^6 \phi(f_n^b) + F_n)) \quad (14)$$

where “ S ” is the segmentation feature output, “ B ” is the boundary feature output, “concat” is the operation of convolving 6 layers of dilation convolution, “ ϕ ” refers to the FC-Relu-FC-Tanh block, “ s ” refers to the segmentation mode, and “ b ” refers to the boundary mode.

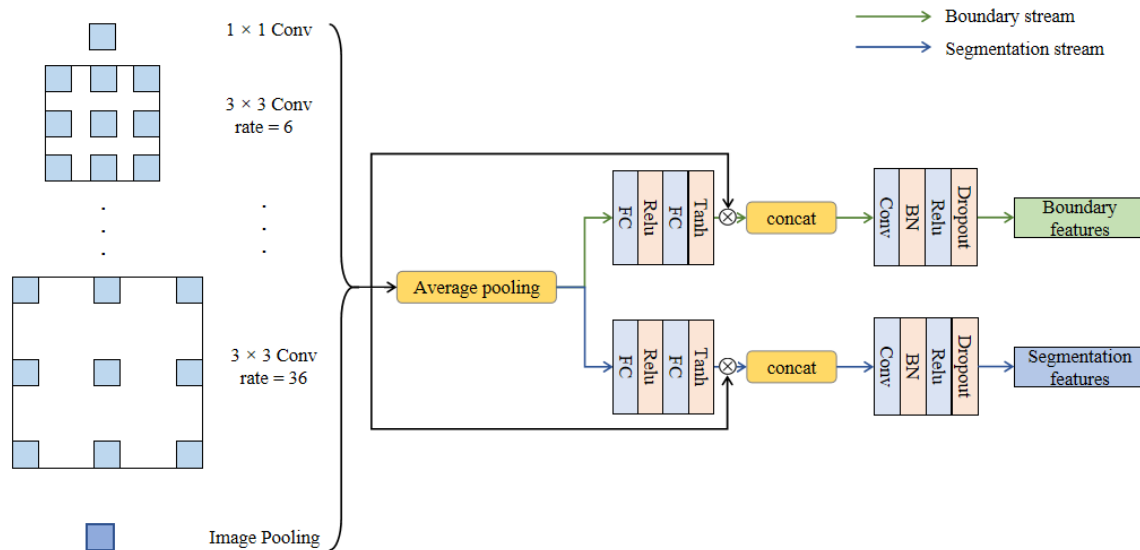


Figure 6. Cross-modal atrous spatial pyramid pooling module structure.

The C-ASPP module processes the features of each MFT layer and outputs the boundary and segmentation features cross-modally. During feature fusion, redundant information is generated owing to differences in the features of different cascades [39,40]. Therefore, they are input into the two lightweight TCHs, as shown in Figure 7; then, attention weights are added to them, and the redundant information is filtered. The green branch represents the boundary flow, and the blue branch represents the segmentation flow. The same attention-weighting formula as in Equations (1)–(4) is used and applied to the boundary and segmentation features, respectively.

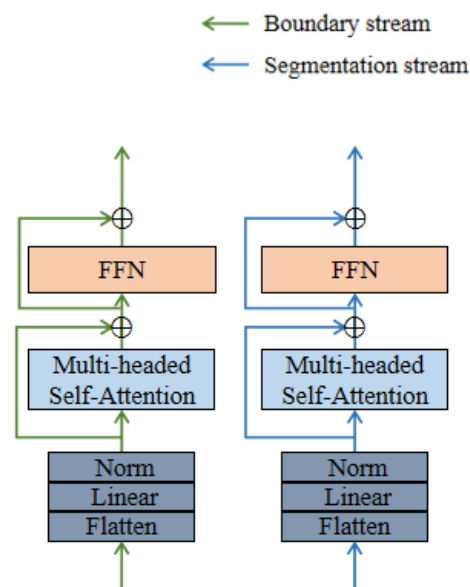


Figure 7. Transformer conversion head.

To realize the contextual linkage of multilevel receptive fields, it is necessary to fuse the boundary features with the segmentation features and the features of the previous layer at each layer. Because the features in the previous layer need to be upsampled for fusion, some features are lost during the upsampling process. To ensure accuracy, it is necessary to perform feature analysis again after fusion; therefore, we designed MHB. As shown in Figure 8, it contains a lightweight ASPP layer and a lightweight transformer header. The segmented features are first fused and enhanced with boundary features, and then passed to ASPP along with the features of the previous layer to achieve context-dependent feature fusion of multilevel receptive fields, which is represented as:

$$F = \sum_{i=1}^6 f_s^i + (f_s^i \otimes f_b^i) \quad (15)$$

“ F ” is the final feature obtained by fusion, “ f_b ” is the boundary feature, “ f_s ” is the segmentation feature, and “ i ” is the number of layers.

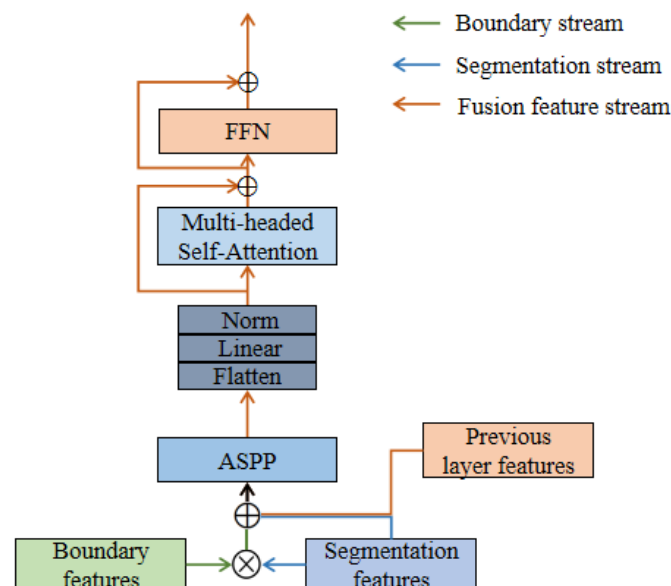


Figure 8. Mixed head block.

The transformer head uses the same principle as TCH to suppress useless information in the fused features, and it uses the attention weighting of the fused features to enhance the features. However, the difference is that the FFN layer reduces the dimensionality of the changed output to adapt the feature dimensions of the latter layer while reducing the number of parameters, and the dimensionality is compressed by a convolution in the last layer to achieve classification.

3.4. Loss Function

In the network framework designed in this paper, segmentation loss and boundary loss are mainly used, where “ G ” is the ground truth. “ G_b ” represents the ground truth of the boundary, “ P ” represents the prediction result, and “ P_b ” represents the boundary prediction result. The segmentation loss and boundary loss can then be expressed as follows, respectively.

$$L_{seg} = 1 - \frac{2|P \cap G| + smooth}{|P| + |G| + smooth} \quad (16)$$

$$L_{boundary} = 1 - \frac{2|P_b \cap G_b| + smooth}{|P_b| + |G_b| + smooth} \quad (17)$$

where the value of “ $smooth$ ” is 1 to prevent the denominator from becoming zero.

4. Experiment

4.1. Dataset and Settings

GDD Dataset [15]: This is a dataset focusing on glass detection and covering various scenarios such as shopping malls, streets, offices, and residences. It contains 2827 indoor images and 1089 outdoor images, and in the same way as for the data division provided by the GDD dataset, we used 2980 of the images as training data and the remaining 936 images as test data.

Implementation Details: First, we binarized the ground truth of the dataset and transformed it into a single-channel image, which facilitated the training process to focus on learning the labels of the target objects. We built our network model using PyTorch 1.8.0 and Cuda 11.3. The platform we used to perform the training was 3 RTX A6000, with an initial learning rate of 1×10^{-4} decayed by the poly strategy [44]. We used a ResNeXt-101 [45] backbone network pretrained on ImageNet [46], and the learning rate decayed linearly to 1×10^{-6} after training. The optimizer used was Adamw, with epsilon set to 1×10^{-8} and weights decaying to 1×10^{-4} . The batch size was set to 12 for each GPU. After training on GDD [15] for 500 epochs, convergence was achieved after an average time of 36 h. The input size for the experiments was 512×512 resolution. For a fair comparison, we did not use any augmented data, online hard example mining (OHEM), auxiliary loss, or class-weighted loss.

4.2. Evaluation Metrics

We followed the work of [20] using four semantic segmentation metrics to evaluate the performance of glass detection: intersection over union (*IoU*), *F*-measure (*F β*) [47], mean absolute error (MAE), and the balance error rate (BER) [48].

The *IoU* is a widely used evaluation tool in the semantic segmentation field and is defined as:

$$IoU = \frac{\sum_{i=1}^H \sum_{j=1}^W (G(i,j) \times P(i,j))}{\sum_{i=1}^H \sum_{j=1}^W (G(i,j) + P(i,j) - G(i,j) \times P(i,j))} \quad (18)$$

where “*H*” and “*W*” are the height and width of the image, “*G*” is the ground truth mask, and “*P*” is the binarized prediction result mask. The glass region is labeled 1, and the other regions are labeled 0.

The *F*-measure is also an evaluation criterion commonly used in the segmentation field to assess the performance of classification models. Following [49], this study uses the weighted *F*-measure [47], which has been proven to be more accurate than the *F*-measure for evaluation results according to some recent studies. It is specifically defined as:

$$F_{\beta}^w = \left(1 + \beta^2\right) \frac{Precision^w \times Recall^w}{\beta^2 \times Precision^w + Recall^w} \quad (19)$$

where β ($\beta = 1$) is the parameter that regulates whether the detection is excessive, “*precision^w*” is the weighable accuracy, which represents a measure of accuracy, and “*Recall^w*” is the weighted recall, representing a measure of completeness.

The MAE is the average of the absolute error between the ground truth and prediction result, which is used in this study in the form of:

$$MAE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W |P(i,j) - G(i,j)| \quad (20)$$

where “*P(i,j)*” denotes the label information of the binary prediction outcome mask on “*(i,j)*”.

The BER evaluates the mean value of the respective prediction error rate in positive and negative case samples and is calculated as:

$$BER = 100 \times \left(1 - \frac{1}{2} \left(\frac{TP}{N_p} + \frac{TN}{N_n}\right)\right) \quad (21)$$

where “ TP ” is the number of pixels predicted to be correct for the target, “ TN ” is the number of pixels predicted to be correct for the nontarget, N_p is the total number of pixels for the target, and N_n is the total number of pixels for the nontarget.

Higher values of IoU and F -measure are desired, whereas lower values of MAE and BER are desired.

4.3. Comparison Methods

We selected 21 methods for comparison with our approach to validate the performance of our TGSNet, based on papers in fields related to our research topic and arranged by year of publication. These included the semantic segmentation methods PSPNet [10], ICNet [50], DeepLab3+ [31], BiSeNet [51], DANet [7], CCNet [52], GFFNet [53], FaPN [54]; salient object detection methods RAS [55], DSS [56], EGNet [36], F3Net [57]; transparent object segmentation methods TransLab [58], Trans2Seg [16], Trans4Trans [35]; mirror segmentation methods MirrorNet [17]; and glass segmentation methods GDNet [15], GSD [59], EBLNet [26], and PGSNet [20]. To fairly compare the performance of the network frameworks, without any augmented data, we used the recommended optimal parameters for the public code papers, and we trained on the GDD dataset. For nonpublic code papers, we used the values provided in a previous study [20]. All the evaluation results were computed using the same evaluation codes.

4.4. Comparison with Existing Methods

In Table 1, we show the evaluation results of our method on the GDD dataset compared against other studies, with specific evaluation results from the aforementioned previous study [20]. As can be seen from the comparison in the table, our method has a much higher IoU score than do the other methods. The IoU metric is the most important performance metric in the field of semantic segmentation, and the IoU of our method is 0.66% higher than that of the current state-of-the-art glass segmentation method PGSNet, and the F_{β}^w of our method is 0.7% higher than PGSNet. As shown in Figure 9, a qualitative comparison was made with six state-of-the-art glass and transparent object segmentation methods (salient object detection method ITSD [60], transparent object segmentation methods Trans2Seg [16] and Trans4Trans [58], mirror segmentation method MirrorNet [17], glass segmentation method EBLNet [26], and GDNet [15]). Our TGSNet can segment glass regions in dark light (rows 2 and 3), in multiple glass regions (rows 4, 5, 6, 8, 9, and 10), and in large glass region (rows 1, 7, and 13), as well as outdoor natural lighting conditions (rows 4, 5, 11, and 14), and the rest are lighting conditions for different types of lights indoors with almost no false detection with almost no false detection and with smoother edges than the other methods while ensuring the integrity of the segmented glass regions. This is mainly because the multiple-use lightweight transformer header in TGSNet can filter redundant information and explore different background levels to enhance the features. For example, on the right side of the resulting image in row 6, other methods would label part of the nontarget region as glass, while our method filters the redundant information contained in the features multiple times due to its different receptive fields. Filtering is carried out both to locate the glass region more accurately while limiting the impact of the background (rows 1 and 7) and to retain more detail (rows 12 and 14).

Table 1. Quantitative comparison of our designed network TGSNet with classical and non-open-source algorithms on the GDD dataset. Red indicates the best results, and blue indicates the second-best results.

Method	Published Journals	Backbone	$IoU \uparrow$	$F_{\beta}^w \uparrow$	GDD [15]	
					$MAE \downarrow$	$BER \downarrow$
PSPNet [10]	CVPR'17	ResNet-50	84.06	0.867	0.084	8.79
ICNet [50]	ECCV'18	ResNet-50	69.59	0.747	0.164	16.10
DeepLab3+ [31]	ECCV'18	ResNet-50	69.95	0.767	0.147	15.49
BiSeNet [51]	ECCV'19	ResNet-50	80.00	0.830	0.106	11.04

Table 1. Cont.

Method	Published Journals	Backbone	IoU \uparrow	$F_w^w \uparrow$	GDD [15]	
					MAE \downarrow	BER \downarrow
DANet [7]	CVPR'19	ResNet-50	84.15	0.864	0.089	8.96
CCNet [52]	ICCV'19	ResNet-50	84.29	0.867	0.085	8.63
GFFNet [53]	AAAI'20	ResNet-50	82.41	0.855	0.090	9.11
FaPN [54]	ICCV'21	ResNet-101	86.65	0.887	0.062	5.69
RAS [55]	ECCV'18	ResNet-50	80.96	0.830	0.106	9.48
DSS [56]	TPAMI'19	ResNet-50	80.24	0.799	0.123	9.73
EGNet [36]	ICCV'19	ResNet-50	85.05	0.870	0.083	7.43
F3Net [57]	AAAI'20	ResNet-50	84.79	0.870	0.082	7.38
ITSD [60]	CVPR'20	ResNet-50	83.72	0.862	0.087	7.77
MirrorNet [17]	ICCV'19	ResNeXt-101	85.07	0.866	0.083	7.67
TransLab [58]	ECCV'20	ResNet-50	81.64	0.849	0.097	9.70
GSD [59]	CVPR'21	ResNeXt-101	87.53	0.895	0.066	5.90
PGSNet [20]	TIP'22	ResNeXt-101	87.81	0.901	0.062	5.56
TGSNet (our)	\	ResNeXt-101	88.47	0.908	0.058	5.70

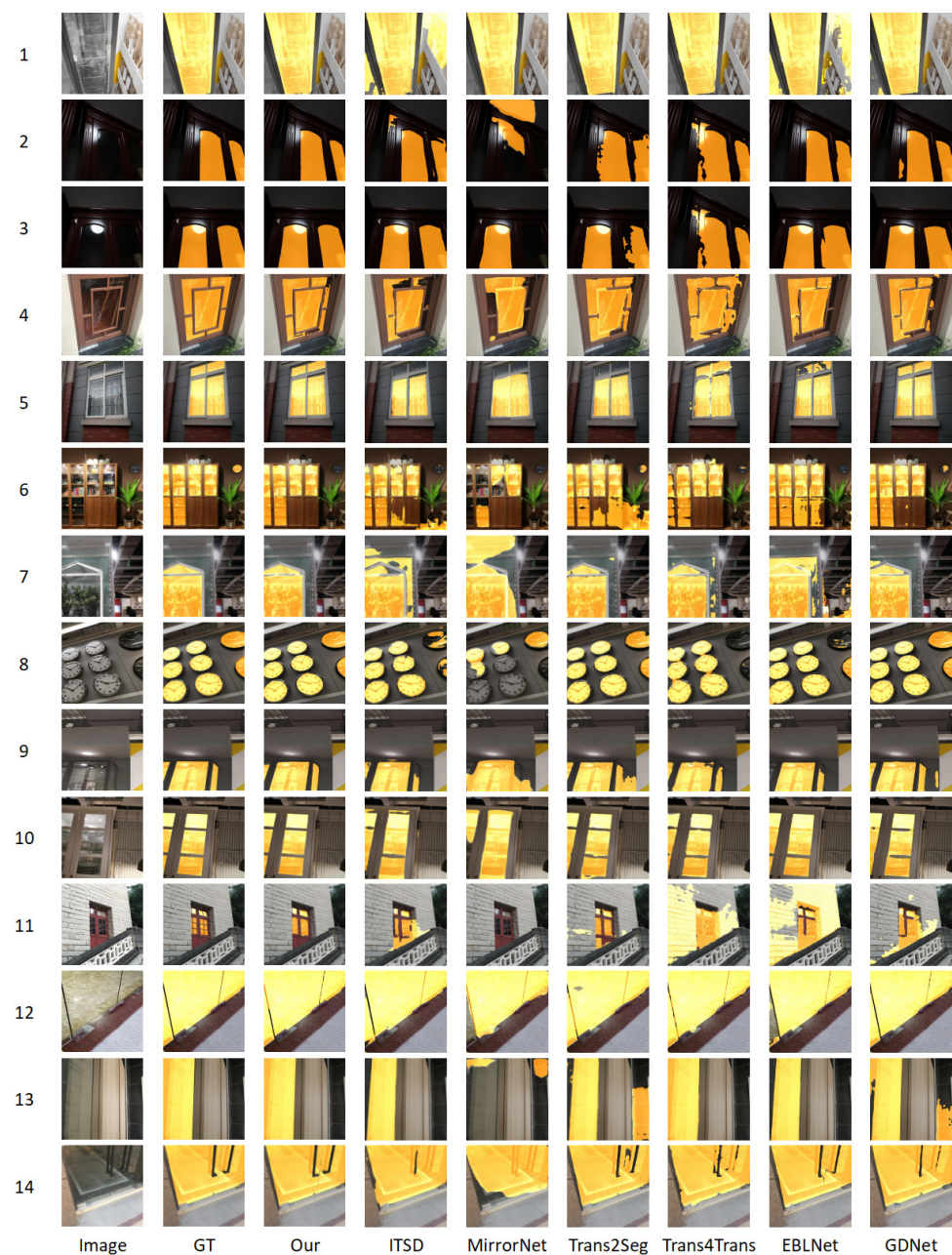


Figure 9. Results of visual comparison between our TGSNet and six other methods.

Since the GDD dataset does not contain test data for lighting conditions, we took photos of glass doors at the same location during the day and night as a small light condition experiment. Figure 10 shows that our method works well for glass segmentation both in the daytime and nighttime, where we made the ground truth manually.

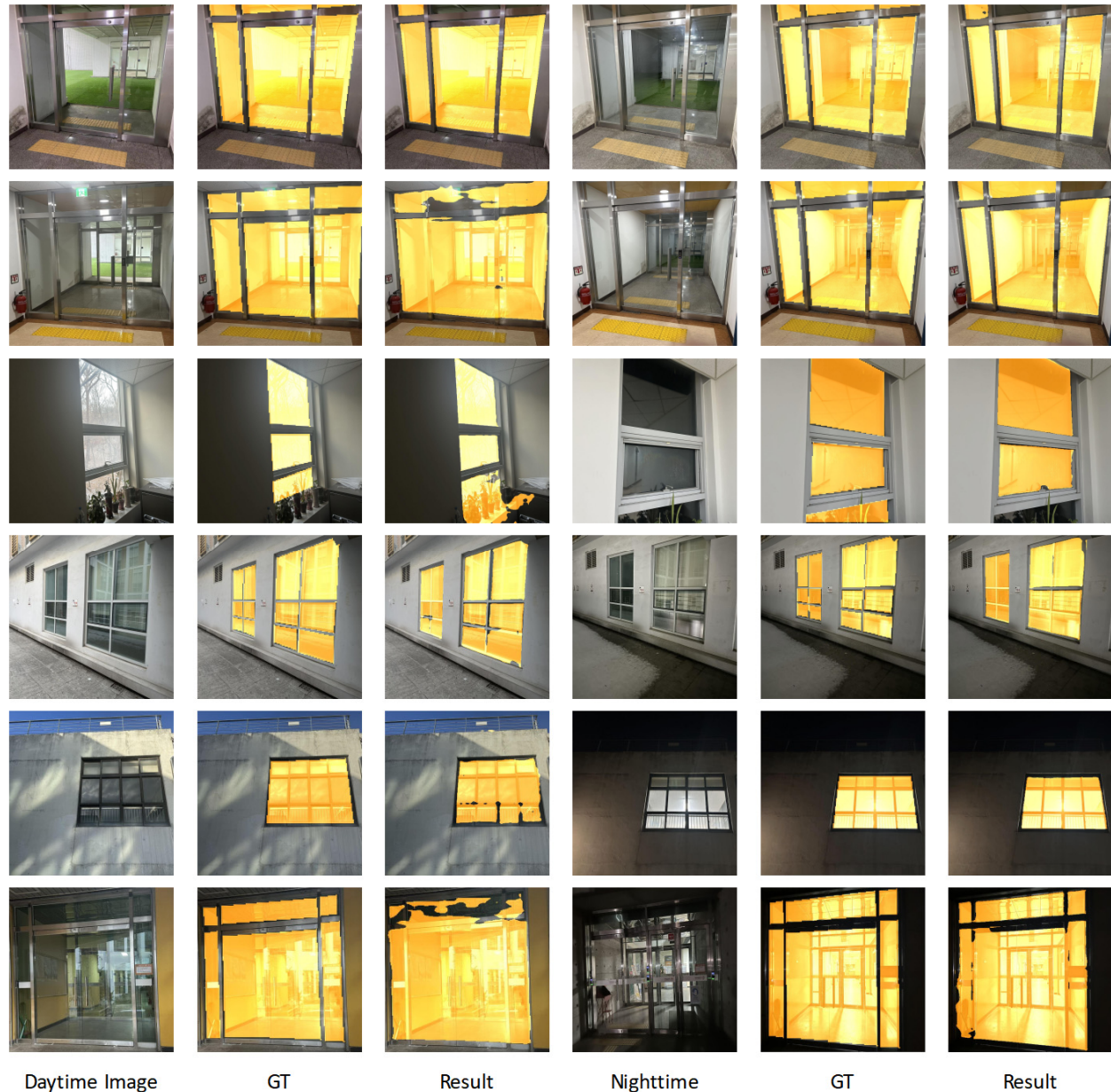


Figure 10. Results under different lighting conditions (day and night).

As shown in Table 2, we retrained some open-source glass and transparent object detection methods strictly according to the parameters provided in the original paper and evaluated them for comparison. In particular, none of the tested methods use any data augmentation or OHEM, auxiliary, or class-weighted loss. We evaluated only the glass segmentation performance. Comparing the values of the results in Table 2, the IoU, which is the most important evaluation metric, of our method is much higher than that of the other methods, thus showing that our method's performance is the best.

Table 2. Comparison results of our algorithm with open-source glass and transparent object segmentation methods on the GDD dataset. The red color marks the best results, and the blue color marks the second-best results.

Method	Published Journals	Backbone	IoU↑	GDD [15]		
				$F_{\beta}^w \uparrow$	MAE↓	BER↓
Trans2seg [16]	IJCAI'21	ResNet-50	84.41	0.872	0.078	7.36
Trans4Trans [35]	ICCVW'21	PVT-Medium	84.94	0.878	0.076	6.86
GDNet [15]	CVPR'20	ResNeXt-101	87.63	0.898	0.063	5.62
EBLNet [26]	CVPR'21	ResNeXt-101	84.98	0.879	0.076	7.24
TGSNet (our)	\	ResNeXt-101	88.47	0.908	0.058	5.70

As shown in Table 3, the results of the computational efficiency as well as the number of parameters, inference speed, and memory usage between our TGSNet and the state-of-the-art glass segmentation methods are compared. Since GSD [59] and PGSNet [20] are not open-source, we can only obtain partial results from paper [20]. The results of all methods are obtained in the same environment using the python library “ptflops” [61]. In terms of computational efficiency, TGSNet requires 40% less FLOPs than PGSNet [20] for each resolution of the input, while it is lower than all other methods. In terms of number of parameters, it is slightly lower than GDNet [15] and higher than EBLNet [26]. The inference speed and memory usage are higher than other open source methods.

Table 3. Comparison results of our algorithm with state-of-the-art glass and transparent object segmentation methods in terms of computational efficiency as well as average inferred speed per image and memory usage. Red indicates the best results, and blue indicates the second-best results.

Methods	FLOPs (G)				MParams	Speed (per Image)	Memory
	352×352	384×384	416×416	512×512			
GDNet [15]	194.48	231.45	271.63	411.46	201.72	0.18 s	1623 MiB
GSD [59]	77.892	92.697	108.790	/	/	/	/
EBLNet [26]	255.34	303.87	356.63	540.2	111.45	0.23 s	6515 MiB
PGSNet [20]	80.789	96.145	112.837	/	/	/	/
TGSNet (our)	49.86	58.98	69.57	104.87	185.472	0.26 s	8921 MiB

4.5. Ablation Experiments

In this section, we describe three sets of ablation experiments. We verified the effectiveness of the CAM, MFT, and CCFF components through different experiments. We compared the performance of the CAM module with CNN in extracting backbone features, the effect of using self-attention versus convolutional attention on the number of parameters in MFT, and the effect of each feature analysis module on the overall performance in CCFF. We present the results in Tables 4–6 and Figures 10–12. In the table, “Networks” represents the network structure used, “Backbone” represents the backbone network used for training, and “MParams” represents the number of parameters.

Table 4. CAM module ablation experimental results: “conv” refers to the extraction of backbone features at different levels using convolutional methods, and “CAM” denotes our contextual attention module.

Networks	Backbone	IoU↑	GDD [15]		
			$F_{\beta}^w \uparrow$	MAE↓	BER↓
a. Conv + MFT + CCFF	ResNeXt-101	87.29	0.898	0.062	6.36
b. CAM + MFT + CCFF	ResNeXt-101	88.47	0.908	0.058	5.70

Table 5. Results of ablation experiments show that using convolutional attention in MFT can effectively reduce the number of parameters without loss of accuracy; “atten” indicates self-multi-headed attention, and “conv atten” indicates multiscale convolutional attention.

Networks	Backbone	MParams	IoU \uparrow	GDD [15]		
				$F_{\beta}^w \uparrow$	MAE \downarrow	BER \downarrow
a. CAM + CCFF	ResNeXt-101	63.260	87.58	0.884	0.074	6.36
b. CAM + self atten +CCFF	ResNeXt-101	249.389	88.28	0.902	0.060	5.89
c. CAM + conv atten +CCFF	ResNeXt-101	185.472	88.47	0.908	0.058	5.70

Table 6. Results of CCFF module ablation experiments; “C-ASPP,” “MHB,” and “TCH,” are the multiscale cross-modal feature extraction, fusion head block, and transformer transformation head in the CCFF module, respectively.

Networks	Backbone	IoU \uparrow	GDD [15]		
			$F_{\beta}^w \uparrow$	MAE \downarrow	BER \downarrow
a. CAM + MFT + C-ASPP	ResNeXt-101	87.31	0.896	0.063	6.58
b. CAM + MFT + C-ASPP + MHB	ResNeXt-101	88.05	0.903	0.060	6.00
c. CAM + MFT + C-ASPP + TCH + MHB	ResNeXt-101	88.47	0.908	0.058	5.70

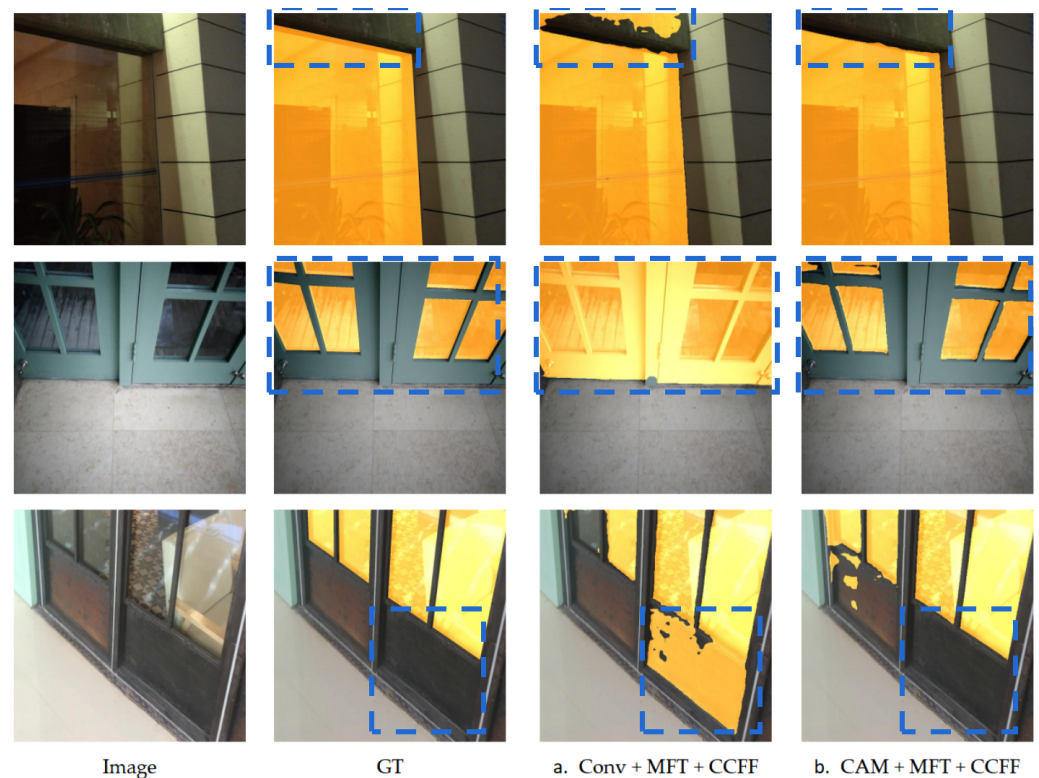


Figure 11. Comparison of results of CAM module ablation experiments. (a) uses the convolution method and (b) uses the CAM module.

4.5.1. Effectiveness of the CAM Module

In this section, to verify the validity of CAM, the convolution method is used for comparison with CAM and to replace the CAM structure and continue the experiment without change in the other structures. The experiment was divided into two parts, (a) and (b). Part (a) uses convolution to obtain backbone features and uses MFT and CCFF, and part (b) uses our CAM to obtain backbone features and also uses MFT and CCFF.

As shown in Figure 11, using our CAM enhances the feature information from the feature backbone while filtering out erroneous features owing to differences between different layers (as shown in the blue dashed box in Figure 11) relative to using a convolution approach to extract the feature backbone and perform feature fusion at different levels.

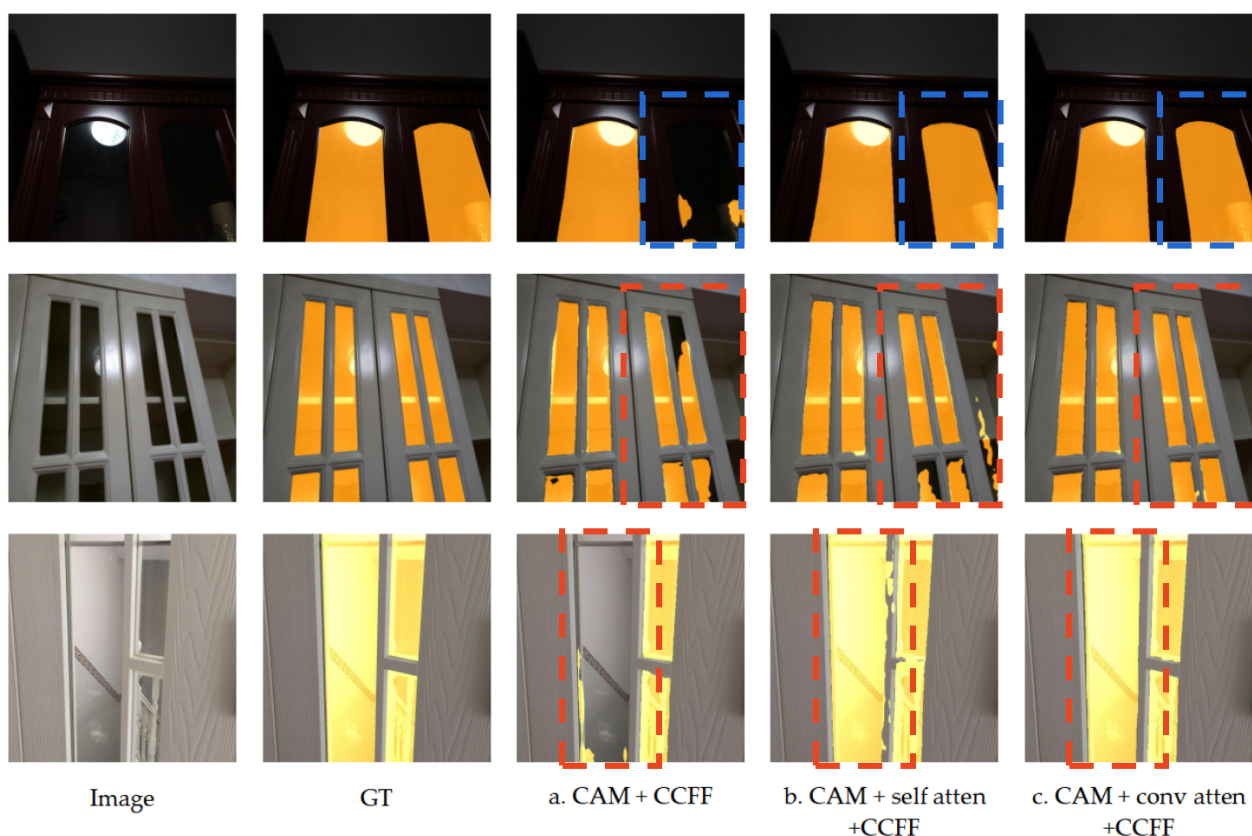


Figure 12. Comparison of the results of CCFF module ablation experiments; (a) is the result of using only C-ASPP, (b) of using C-ASPP and MHB, and (c) of using C-ASPP + TCH + MHB.

In Table 4, we quantitatively compare the evaluation results of the two approaches and clearly demonstrate that our CAM module is more effective for backbone feature extraction and fusion.

4.5.2. Effectiveness of the MFT Module

In this section, we discuss the case where a structure other than the MFT is guaranteed to be constant. We introduced the self-attention method and the multiscale convolutional attention method into MFT separately to compare them quantitatively with the case where MFT was not used. The aforementioned experiments were conducted to verify the validity of the MFT. This part is divided into groups (a), (b), and (c) for the experiments, where (a), (b), and (c) all use the same CAM and CCFF; the difference is that (a) does not use the MFT module, (b) uses MFT with multi-headed self-attention, and (c) uses MFT with multiscale convolutional attention.

As shown in Table 5, it can be found that the total number of parameters significantly declined after using the multiscale convolutional attention mechanism compared with using the multi-headed attention mechanism. Simultaneously, the accuracy of segmentation was ensured, and none of the four evaluation metrics decreased. It can be noted that MFT improves the IoU value by 0.19% after using multiscale convolutional attention is used relative to using self-attention, which proves that the feature extraction performance is improved.

In addition, as shown in Figure 12, we qualitatively show that a more complete detection of the glass region is possible with the introduction of the multi-head attention mechanism and the multiscale convolutional attention mechanism (as shown in the blue dashed box in row 1) and can segment the target and nontarget regions more precisely, preserving greater detail (red-dashed boxes in b and c).

4.5.3. Effectiveness of the CCFF Module

The purpose of this section is to verify the effectiveness of each module in the CCFF by comparing the results of using (a) only C-ASPP, (b) using C-ASPP and TCH, and (c) using C-ASPP, TCH, and MHB, while keeping other conditions constant. In the cross-modal context feature fusion phase, C-ASPP is our basic multiscale cross-modal semantic feature extraction module, and TCH aims to reweight the feature information and plays a role equivalent to that of the transformer header in CAM. Because there is redundant information when features are fused at different levels because of the differences between feature information at different levels [39,40], feature fusion using only superposition will produce performance limitations; therefore, we designed MHB to further enhance the fused features. By performing another multiscale feature extraction at each fusion stage, the effect of feature enhancement is achieved, and better results are obtained when MHB is introduced to further filter the features (Figure 13).

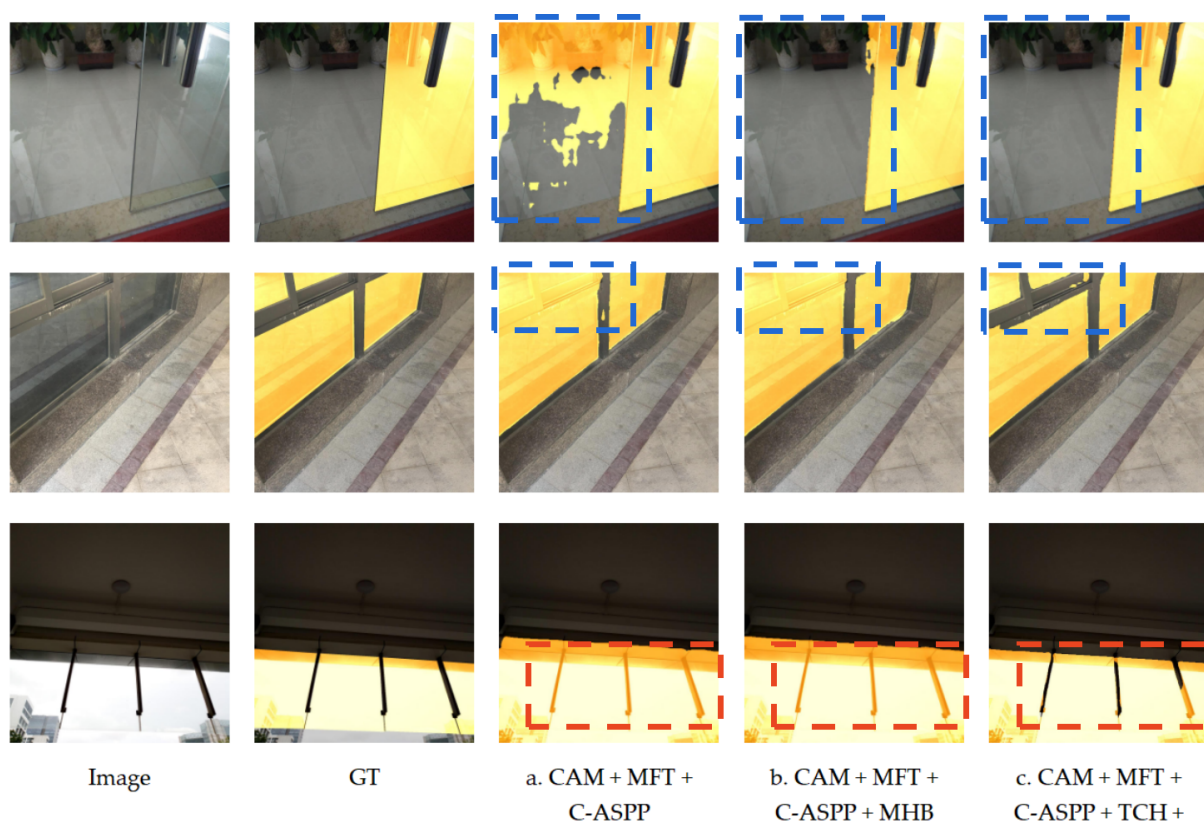


Figure 13. Comparison of experimental results of multiscale convolutional attention ablation; (a) does not use MFT, (b) uses self-multi-headed attention, and (c) uses multi-scale convolutional attention.

We quantitatively tested the results after using only C-ASPP and gradually introducing MHB and TCH. The results are shown in Table 6 and demonstrate that our CCFF module plays an important role in improving the glass detection performance after MHB and TCH are introduced.

In addition, we performed a qualitative analysis, as shown in Figure 13, which showed that the enhancement of fused features using MHB can filter large error areas (blue dashed box in row 1), whereas the algorithm that optimizes the features before introducing TCH to fuse them makes the results more accurate in classifying nontarget regions (red dashed box in row 3). This indicates that the CCFF module we built for fusing features at different layers filters redundant information and enhances the details, which promotes the optimization of the overall framework performance.

5. Conclusions

We challenged the glass segmentation task and designed a framework for transformer glass detection, called TGSNet. We constructed a CAM to extract backbone features using a parallel transformer head structure. We also designed MFT using convolutional attention to obtain more detailed feature information by further analyzing the features from different receptive fields. Our CCFF module aims to improve the overall performance of the network through flexible use of the transformer header for augmenting features and concatenating feature information across layers. All these structures have improved in terms of performance of feature extraction. The experimental results show that our method achieves state-of-the-art performance on the GDD dataset and achieves good performance in experiments under different lighting conditions. In the future, we plan to try to experiment on more datasets or build a dataset for light conditions for experimentation and extend our approach to other areas, for example, to the video detection of glass, removing the noise generated by glass in point clouds generated from 2D images, and further enhancing the ability to detect glass in outdoor scenes as well as in ultra-high-resolution images.

Author Contributions: Conceptualization, X.H.; Formal analysis, X.H.; Funding acquisition, K.C.; Investigation, R.G.; Methodology, X.H.; Project administration, K.C.; Software, X.H.; Supervision, K.C.; Validation, X.H. and R.G.; Visualization, X.H.; Writing—original draft, X.H.; Writing—review and editing, R.G., S.Y. and K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (22ZH1200, The research of the fundamental media contents technologies for hyper-realistic media space) and by the Dongguk University Research Fund of 2022 (S-2022-G0001-00136).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: URL: https://mhaiyang.github.io/CVPR2020_GDNet/index.html (accessed on 2 February 2023) [15].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gao, R.; Li, M.; Yang, S.J. Reflective Noise Filtering of Large-Scale Point Cloud Using Transformer. *Remote Sens.* **2022**, *14*, 577. [CrossRef]
2. Liu, F.; Shen, C.; Lin, G. Deep convolutional neural fields for depth estimation from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5162–5170.
3. Zheng, C.; Cham, T.J.; Cai, J. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 767–783.
4. Zhang, L.; Dai, J.; Lu, H. A bi-directional message passing model for salient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1741–1750.
5. Wu, Z.; Su, L.; Huang, Q. Cascaded partial decoder for fast and accurate salient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3907–3916.
6. Liu, J.J.; Hou, Q.; Cheng, M.M. A simple pooling-based design for real-time salient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3917–3926.
7. Fu, J.; Liu, J.; Tian, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3146–3154.
8. He, K.; Gkioxari, G.; Dollár, P. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
9. Dai, J.; Li, Y.; He, K. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Annual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, 5–10 December 2016.
10. Zhao, H.; Shi, J.; Qi, X. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.

11. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 568–578.
12. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
13. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
14. Zhang, H.; Dana, K.; Shi, J. Context encoding for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7151–7160.
15. Mei, H.; Yang, X.; Wang, Y. Don't hit me! glass detection in real-world scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3687–3696.
16. Xie, E.; Wang, W.; Wang, W. Segmenting Transparent Objects in the Wild with Transformer. *arXiv* **2021**, arXiv:2101.08461.
17. Yang, X.; Mei, H.; Xu, K. Where is my mirror? In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8809–8818.
18. Chen, L.C.; Papandreou, G.; Schroff, F. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
19. Cao, Y.; Zhang, Z.; Xie, E. FakeMix augmentation improves transparent object detection. *arXiv* **2021**, arXiv:2103.13279.
20. Yu, L.; Mei, H.; Dong, W. Progressive Glass Segmentation. *IEEE Trans. Image Process.* **2022**, *31*, 2920–2933. [[CrossRef](#)] [[PubMed](#)]
21. Huynh, C.; Tran, A.T.; Luu, K. Progressive semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 16755–16764.
22. Zhao, J.X.; Liu, J.J.; Fan, D.P. EGNNet: Edge guidance network for salient object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8779–8788.
23. Huo, D.; Wang, J.; Qian, Y. Glass Segmentation with RGB-Thermal Image Pairs. *arXiv* **2022**, arXiv:2204.05453.
24. Lin, J.; Yeung, Y.H.; Lau, R.W.H. Depth-aware glass surface detection with cross-modal context mining. *arXiv* **2022**, arXiv:2206.11250.
25. Mei, H.; Dong, B.; Dong, W. Glass Segmentation Using Intensity and Spectral Polarization Cues. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 12622–12631.
26. He, H.; Li, X.; Cheng, G. Enhanced boundary learning for glass-like object segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 15859–15868.
27. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
28. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015; pp. 234–241.
29. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)] [[PubMed](#)]
30. Chen, L.C.; Papandreou, G.; Kokkinos, I. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [[CrossRef](#)] [[PubMed](#)]
31. Chen, L.C.; Zhu, Y.; Papandreou, G. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
32. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1440–1448.
33. Wang, W.; Xie, E.; Li, X. Pvt v2: Improved baselines with pyramid vision transformer. *Comput. Vis. Media* **2022**, *8*, 415–424. [[CrossRef](#)]
34. Guo, M.H.; Lu, C.Z.; Hou, Q. SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation. *arXiv* **2022**, arXiv:2209.08575.
35. Zhang, J.; Yang, K.; Constantinescu, A. Trans4Trans: Efficient transformer for transparent object segmentation to help visually impaired people navigate in the real world. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 1760–1770.
36. Merget, D.; Rock, M.; Rigoll, G. Robust facial landmark detection via a fully-convolutional local-global context network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 781–790.
37. Aboutaleb, H.; Pavlova, M.; Gunraj, H. MEDUSA: Multi-Scale Encoder-Decoder Self-Attention Deep Neural Network Architecture for Medical Image Analysis. *Front. Med.* **2021**, *8*, 2891. [[CrossRef](#)] [[PubMed](#)]
38. Vaswani, A.; Shazeer, N.; Parmar, N. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; p. 30.
39. Xiao, J.; Zhao, T.; Yao, Y. Context augmentation and feature refinement network for tiny object detection. In Proceedings of the ICLR 2022 Conference, Virtual, 25–29 April 2022.
40. Chen, L.I.; Jianxun, L.I. Orthogonal Features Extraction Method and Its Application in Convolution Neural Network. *J. Shanghai Jiaotong Univ.* **2021**, *55*, 1320.

41. Zhou, B.; Khosla, A.; Lapedriza, A. Object detectors emerge in deep scene cnns. *arXiv* **2014**, arXiv:1412.6856.
42. Peng, C.; Zhang, X.; Yu, G. Large kernel matters—Improve semantic segmentation by global convolutional network. In Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4353–4361.
43. Ding, X.; Guo, Y.; Ding, G. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1911–1920.
44. Liu, W.; Rabinovich, A.; Berg, A.C. Parsenet: Looking wider to see better. *arXiv* **2015**, arXiv:1506.04579.
45. Xie, S.; Girshick, R.; Dollár, P. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
46. Deng, J.; Dong, W.; Socher, R. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
47. Margolin, R.; Zelnik-Manor, L.; Tal, A. How to evaluate foreground maps? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 248–255.
48. Nguyen, V.; Yago Vicente, T.F.; Zhao, M.; Hoai, M.; Samaras, D. Shadow detection with conditional generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4510–4518.
49. Fan, D.P.; Cheng, M.M.; Liu, Y. Structure-measure: A new way to evaluate foreground maps. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4548–4557.
50. Zhao, H.; Qi, X.; Shen, X. Icnet for real-time semantic segmentation on high-resolution images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 405–420.
51. Yu, C.; Wang, J.; Peng, C. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 325–341.
52. Huang, Z.; Wang, X.; Huang, L. Ccnet: Criss-cross attention for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 603–612.
53. Li, X.; Zhao, H.; Han, L. Gated fully fusion for semantic segmentation. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 11418–11425. [[CrossRef](#)]
54. Huang, S.; Lu, Z.; Cheng, R. FaPN: Feature-aligned pyramid network for dense image prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 864–873.
55. Chen, S.; Tan, X.; Wang, B. Reverse attention for salient object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 234–250.
56. Hou, Q.; Cheng, M.M.; Hu, X. Deeply supervised salient object detection with short connections. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3203–3212.
57. Wei, J.; Wang, S.; Huang, Q. F³Net: Fusion, feedback and focus for salient object detection. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12321–12328.
58. Xie, E.; Wang, W.; Wang, W. Segmenting transparent objects in the wild. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 696–711.
59. Lin, J.; He, Z.; Lau, R.W.H. Rich context aggregation with reflection prior for glass surface detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 13415–13424.
60. Zhou, H.; Xie, X.; Lai, J.H. Interactive two-stream decoder for accurate and fast saliency detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9141–9150.
61. Available online: <https://github.com/sovrasov/flops-counter.pytorch> (accessed on 2 February 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.