Utilizing Cross Layer Attentions for Semantic Segmentation of Small Objects

Yu-Hsien Chung, Chi-Hsuan Lu, Jung-Hui Cho and Chih-Chang Yu
Chung Yuan Christian University, Taoyuan, Taiwan
E-mail: ccyu@cycu.edu.tw Tel: +886-2659999

***Abstract*— Segmenting small objects poses significant challenges in semantic segmentation tasks. This is because the downsampling process employed by many current deep learning networks tends to suppress the features of small objects, making them difficult to be retained in the encoded latent space. To mitigate this issue, we propose the Cross Layer Attention (CLA) encoder/decoder block, which aims to correlate informative features between deep feature layers. CLA calculates the correlation across different feature layers in the encoder and then subsequently integrating this information is integrated into the Feed-Forward Network (FFN) decoder, thus forming a dual decoding process. The designed CLA is able to preserve the features of objects at different positions, so that it can be used to enhance the feature of a small object during the upsampling process. The proposed CLA is experimented with the ADE20K public dataset using the Segformer – B0 and B1 as the backbone network, and the results are imrpoved by 1.37% and 2.38% in mean Intersection over Union (mIoU) compared to the original SegFormer – B1 and B0 model respectively. Furthermore, for objects which often appear smaller than 32×32 pixels in the validation set, the mIoU is also improved by 3.31% and 7.02%. The experiments demonstrates that the proposed CLA enhances the overall semantic segmentation results, particularly for small objects.**

I. INTRODUCTION

In an era of rapid technological advancement, computer vision has gained significant attention across various fields, demonstrating its applications in numerous areas. Generally, Convolutional Neural Networks (CNN) have been widely used in computer vision applications [1]. Most Convolutional Neural Networks (CNNs) utilize convolutional kernels with restricted sizes, thereby the networks have strong capabilities for analyzing local information within a scope. However, this characteristic also poses a limitation, obstructing CNNs from capturing long-range dependencies, such as the correlation between trucks in a distance and nearby cars, which leads to suboptimal performance, including imprecise segmentation outcomes. Although the receptive field can be expanded through the downsampling of feature maps, this process often results in the suppression or even loss of features pertinent to small objects. Consequently, while CNNs exhibit strong performance on larger objects, their efficacy diminishes significantly when dealing with smaller objects. To capture multi-scale features effectively, Lin et al. [2] introduced the Feature Pyramid Network (FPN). FPN uses multi-scale features to enhance the detection of small objects. However, the most models employing FPN remain CNN-based, thereby lacking the capability to capture long-range and cross layer dependencies.

Recently, the transformer architecture [3] has been adopted in computer vision. Unlike CNNs, which use limited-size convolutional kernels, transformers divide an image into several patches and compute self-attentions among patches. The self-attention mechanism allows each patch to attend to all other patches, eliminating information discontinuities and capturing long-range dependencies with global information, thus reducing information loss. However, while transformers capture long-range dependencies, the features of small objects tend to be sparse, and the computation of self-attention requires significant computational resources.

Addressing the difficulty of small object detection requires large models for training. These models often need expensive hardware, posing challenges for practical applications. Therefore, this study aims to improve small object segmentation within the constraints of affordable hardware (e.g., a consumer GPU). The proposed method utilizes the cross-layer attentions (CLA), which are computed in the encoder and store the correlation between different feature maps. In the decoder, CLA is then used to enhance segmentation performance.

The contributions of this study are as follows:

1. A novel CLA encoder/decoder block is introduced. The encoder blocks obtain correlations between different feature maps, while the decoder blocks use the encoder outputs to create upsampling branches, compensating for features lost during encoding process.
2. CLA requires only a small increase in parameters but enhances small object segmentation capabilities, particularly for lightweight models.
3. CLA can be applied to any model with a symmetric encoder-decoder structure.

II. Related Work

In this section, we will briefly describe the backbone network used in this study.

A. Small object segmentation

The segmentation of small objects is challenging due to their small size, which makes their features less prominent. In recent solutions for small object detection, multi-scale feature learning and Generative Adversarial Networks (GAN)-based detectors are widely adopted. For instance, Li et al. [4] proposed SNIP, which employs multiple feature maps at different scales. It filters out unsuitable targets based on the actual size of the objects in the current scale, thereby avoiding the search for small objects in large-scale feature maps. Additionally, Singh and Davis [5] introduced TridentNet, which divides the network into three parallel branches and uses dilated convolutions with different rates to generate feature maps. These feature maps are then merged to generate the final output. The use of parallel branches allows the model to have feature maps with varying receptive field sizes, thereby effectively identify objects of different scales. Although these methods effectively integrate multi-scale feature maps, they still face the challenge that the failure in the module responsible for detecting small objects results in the miss of small objects in the final results.

As GAN-based Detectors, Rabbi et al. [6] proposed a resolution-enhancing model. By enhancing the resolution, more features are generated to improve the recognition rate of small objects. Similar to the previous method, this approach also requires a significant amount of computational resources and large datasets to train the model.

B. Across Feature Map Attention

The Across Feature Map Attention (AFMA) method proposed by Sang et al. [7] is an attachable model that does not require additional training. Furthermore, this method can be applied to most models, such as U-Net [8], DeepLabV3 [9], FPN, and others. The main steps of AFMA are to find the correlation between the original image and a certain feature map in regard to the number of classes. AFMA processes the original image alongside a specific feature map to acquire the attention related to an object between the image and feature map. Subsequently, the decoder output is multiplied by these attention maps for enhancing the features of small objects.

C. SegFormer

The self-attention mechanism of the Transformer model allows the model to obtain image features more globally. SegFormer [10] is a transformer-based framework primarily used for semantic segmentation. Unlike the Vision Transformer (ViT) [11] [12], which uses position encoding or conditional position encoding to obtain positional information, SegFormer considers position encoding redundant for semantic segmentation. Therefore, SegFormer uses Mix-FFN, which directly extracts and utilizes positional information through 3×3 convolution layers and a multilayer MLP. In addition, to reduce the computation cost of transformer models, SegFormer proposed an efficient self-attention module. First it reshapes the input head tensor K from N×C to N/R×(R×C), where N is the size of the image and R is the reduction ratio. Then, a linear layer is used to change its dimension from R×C to C. This process can significantly reduce the computation cost of self-attention. With the use of efficient self-attention, Mix-FFN, and overlap patch merging, SegFormer distinguishes itself from other Transformer models and enhance both accuracy and efficiency.

Upon deploying SegFormer on several public datasets, it was observed that SegFormer still has some room for improving its performance in segmenting small objects. Although AFMA can address this problem, however, AFMA is restricted to be applied to the decoder output. This constraint requires the number of channels in AFMA to match the number of classes, which causes some information loss. Therefore, inspired by AFMA, this study proposed CLA as a method to address this limitation, which will be introduced in the next section.

III. Proposed Method

The proposed network with the CLA module is illustrated in Fig. 1. First, the feature maps generated by the SegFormer encoder are processed using the CLA encoder to capture the correlation across different layers. The decoder block accepts three inputs that contains a concatenated feature map, the CLA and another feature map. The final segmentation results are obtained through another multi-layer perceptron (MLP).

The key difference between the CLA encoder and the AFMA encoder is that AFMA only calculates the correlation of objects of the same class between the original image and the feature maps in a specific layer. In contrast, our model can compute the correlations between any two layers. Therefore, the model can have multiple CLA encoders to preserve the features of small objects to the greatest extent. Another benefit is that we can apply the results of multiple CLA encoders during the decoding process, ensuring that the model can compensate for the lost small object features during upscaling.

A. Encoder

The proposed CLA encoder is designed as a plug-in for the original network encoder. The process is shown in (1). First, a higher-resolution feature map, denoted as F_i, with size $H_i \times W_i \times C_i$ is passed through a 1×1 convolution to change the channel of the feature map to align with the channel size of a lower-resolution feature map, F_j, whose size is $H_j \times W_j \times C_j$. H_i is larger than H_j and W_i is larger than W_j. After this process, we can obtain another feature map whose size is $H_i \times W_i \times C_j$. Next,

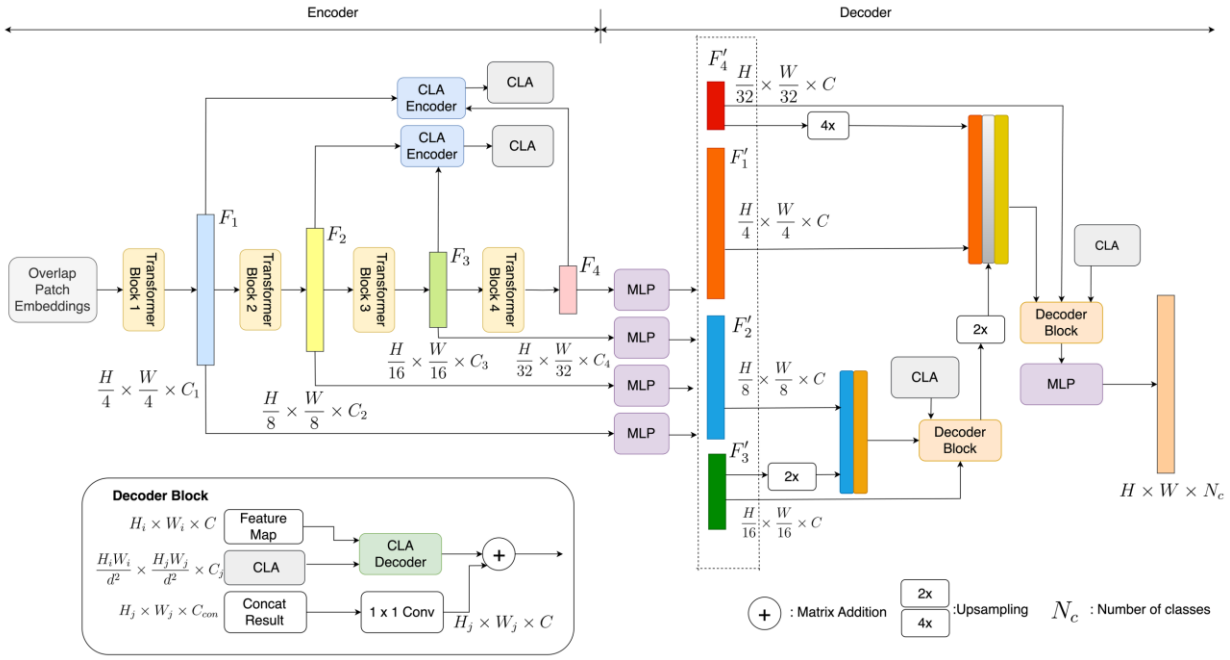

Fig. 1. The model structure proposed in this study. The cross attentions between two feature maps are computed through two CLA encoders. The output of corresponding encoders is then used in the decoder to compensate the feature of small objects.

both feature maps with channel size C_j undergo image patch partitioning (ϕ) with size d^2, changing their sizes to $(H_i \times W_i/d^2) \times d^2 \times C_j$ and $(H_j \times W_j/d^2) \times d^2 \times C_j$, respectively. Finally, we multiply and concatenate them to obtain the CLA_{ij} with size $(H_j \times W_j/d^2) \times (H_j \times W_j/d^2) \times C_j$.

$$\mathrm{CLA}_{ij} = \phi(\mathrm{Conv}_{1\times1}(F_i), d^2) \times \phi(F_j, d^2) \qquad (1)$$

In the original SegFormer, feature extraction is first performed through the transformer blocks, obtaining four different sizes of feature maps. We attach a CLA encoder between two transformer blocks to calculate the attentions. This study uses two different CLA encoders, CLA_{14} and CLA_{23}, which correspond to the use of F_1 with F_4 and F_2 with F_3 respectively.

B. *Decoder Block*

A decoder Block consists of a CLA decoder and a 1×1 convolution. To accommodate the CLA computation, CLA is first passed through a 1×1 convolution for matching the number of channels of the feature map, then the feature map with size $H_j \times W_j \times C$ also undergoes image patch partitioning (ϕ) with size d^2, changing its shape to $(H_j \times W_j/d^2) \times d^2 \times C$. Then it is multiplied by the CLA acquired from the encoder so that the shape of the CLA decoder output O_{ij} goes back to $(H_i \times W_i/d^2) \times d^2 \times C$, which contains the influence of large objects on small objects. The CLA Decoder operation is described in (2).

$$O = \phi(\mathrm{Conv}_{1\times1}(\mathrm{CLA}_{ij}), d^2) \times \mathrm{Linear}(F'_j) \qquad (2)$$

Unlike the original SegFormer, we devise this decoder block which leverages the output of the CLA Decoder. First, given two feature maps F_i and F_j, both through an MLP, the result is denoted as F_i' and F_j'. After that F_j' goes through two paths, one is being upscaled to the size of F_i' and concatenated with F_i'. The output O_{ij} of the previous decoder is also been concatenated with F_i' and F_j' if there exists one. If F_j' is the smallest feature map in the decoder, that block only takes the concatenated result of F_i' and F_j'. For example, in Fig. 2, the first concatenation is F_2' with F_3', then a decoder block is applied. After that, the result is concatenated with F_1' and upsampled to the same size as F_4', then another decoder block is applied. The concatenated result goes through a 1×1 convolution in order to align the channel size with the CLA decoder output. Finally, an MLP is used to fuse all the features to obtain the final segmentation result.

IV. Experimental Results

The experiments in this study were conducted on a Intel core-i9 CPU and an NVIDIA GeForce RTX 4090 GPU using the ADE20K [13] dataset. The ADE20K dataset contains 20,210 images with 150 categories in various daily life scenes, including objects such as buildings, vehicles, furniture, and backgrounds such as sky, roads, and ground. We use SegFormer – B1 and B0 as the network backbone and trained the model for 320k iterations with a batch size of 8.

To demonstrate the performance of the proposed CLA, this study compares the proposed method with the SegFormer. We use two CLA encoders, denoted as CLA_{14} and CLA_{23}. Since the focus of this study is small objects, we also compute the mIoU of small objects, denoted as mIoU@small. We define objects that appear more than 60% of the time in the validation set with a size of less than 32×32 pixels as small objects. As a result, 12 categories are marked as small objects. Table 1 to 3 tabulates the performance with and without the proposed CLA module. From Table 1, we can see that the SegFormer – B1 with CLA module performs better than that without CLA module both in mIoU@small and mIoU. However, the performance is not as good as SegFormer – B2. This is because our model has almost half the number of parameters compared to SegFormer B2, the performance degradation is not surprised. Comparded to SegFormer – B1, with the CLA module, the number of parameters is only increased about 1%, indicating that our proposed technique imposes almost no burden on the model. For overall mIoU, our model improves 1.37% compared to SegFormer – B1. Regarding small objects, the mIoU@small is improved by 3.31%. We also applied our method on Segformer – B0, and the integration of the CLA module resulted in a 2.38% improvement in mIoU and a 7.02% imrpovement in mIoU@small, while only a 5.04% increase in the number of parameters.

Table 2 lists some of the categories improve the most from small objects in the experimental dataset. Compared to SegFromer – B1, the SegFormer – B1 with CLA model improves the mIoU for small objects such as clock, light, ashcan and glass with 16.42%, 4.86%, 4.36%, and 4.09% respectively. Table 3 lists four of the categories improve the most from non-small objects: ship, microwave, barrel and booth. The improvements in terms of mIoU are 41.80%, 24.93%, 18.06%, and 14.69%, respectively. Moreover, Table 2 and Table 3 indicate that the performance of SegFormer – B0 with CLA is still better than SegFormer – B0. These results demonstrate that our method not only improves the performance for non-small objects but also significantly enhances the performance for small objects.

We provide some visual comparisons of small objects (Fig. 2 (a) to (d)) and non-small objects (Fig. 3 (a) to (d)) between SegFormer – B1 and SegFormer – B1 with CLA. Considering small objects, as illustrated in Fig.2 (a), it is evident that the original SegFormer does not perform well in detecting distant skiers dressed in white, resulting in a smaller segmented region compared to the proposed model. This discrepancy is even more obvious in Fig.2 (b), where the number of distant small boats segmented by the original SegFormer is significantly fewer than those segmented by SegFormer with CLA. Fig. 2 (c) also exhibits a noticeable difference; the original SegFormer fails to segment distant small houses, whereas our model successfully segments both small houses. In Fig. 2 (d), part of a distant truck segmented by the original SegFormer is classified as a car, while our model clearly distinguishes different categories, and the segmented truck area is larger in our model compared to the original SegFormer. From the examples mentioned above, it can be observed that SegFormer with CLA performs better in handling small objects.

Table 1 Performance comparisons of models in different scales and the propsosed method

models	Parameter	mIoU@small	mIoU
SegFormer – B0	3.77M	22.64	36.79
SegFormer – B1	13.7M	31.63	41.75
SegFormer – B2	27.5M	35.69	46.33
SegFormer – B0 + CLA	3.96M	29.66	39.17
SegFormer – B1 + CLA	13.8M	34.94	43.12

Table 2 Performance comparison of models with CLA versus those without CLA for small objects.

models	clock	light	ashcan	glass
SegFormer – B0	14.42	35.02	18.98	3.39
SegFormer – B1	16.33	43.26	29.78	5.72
SegFormer – B0 + CLA	24.99	41.23	22.60	5.36
SegFormer – B1 + CLA	32.75	48.12	34.14	9.81

Table 3 Performance comparison of models with CLA versus those without CLA for non-small objects

models	ship	microwave	barrel	booth
SegFormer – B0	54.31	24.91	28.40	15.48
SegFormer – B1	24.51	32.76	27.39	36.96
SegFormer – B0 + CLA	56.22	28.15	44.98	54.25
SegFormer – B1 + CLA	66.31	57.69	45.45	51.65

As shown in Fig. 3 (a), although both the original SegFormer and our model segment two lamps, it is evident that our results are more precise than the original SegFormer. In addition, for large objects such as helicopters and houses (see Fig. 3 (b) to (d)), the proposed method has less errors. To sum up, our method not only improves the ability of segmentation small object, but also non-small object.

V. CONCLUSIONS

This study aims to improve the semantic segmentation of small objects in natural scenes by using cross layer attentions, CLA, to obtain object information across different feature maps, thereby compensating for the features lost during downsampling. The extracted CLA is then utilized in the

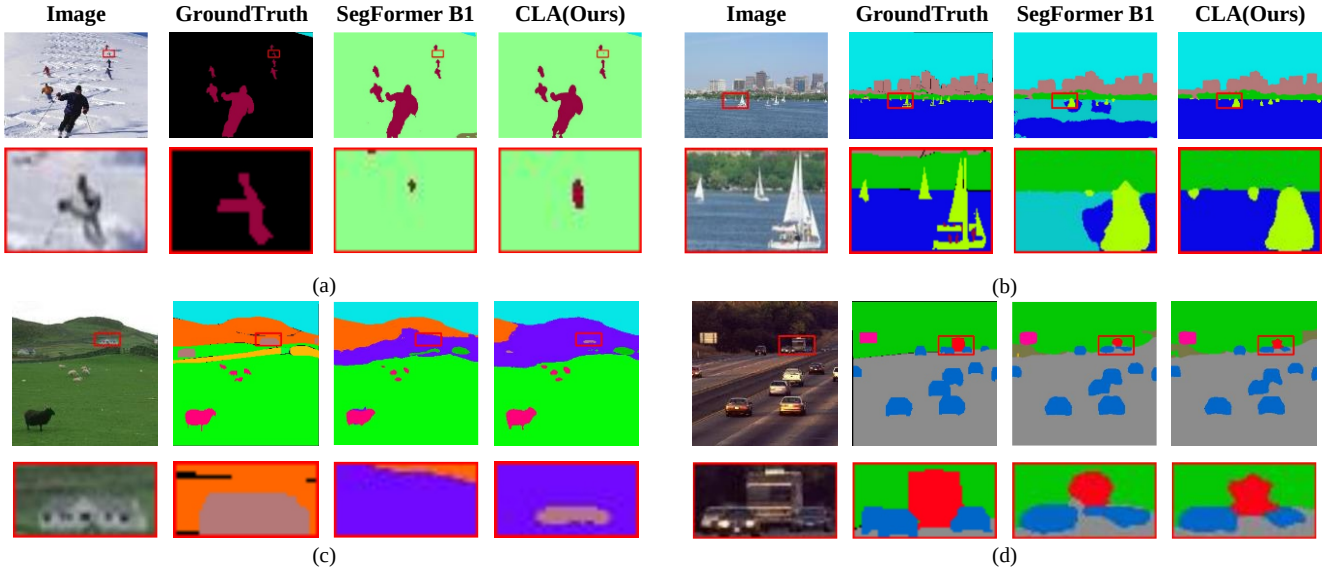

Fig. 2. Visual comparison of small objects. Each column from left to right in a subfigure represents the original image, ground truth, SegFormer B1 segmentation results, and our method. (a) person (b) boat (c) building (d) truck

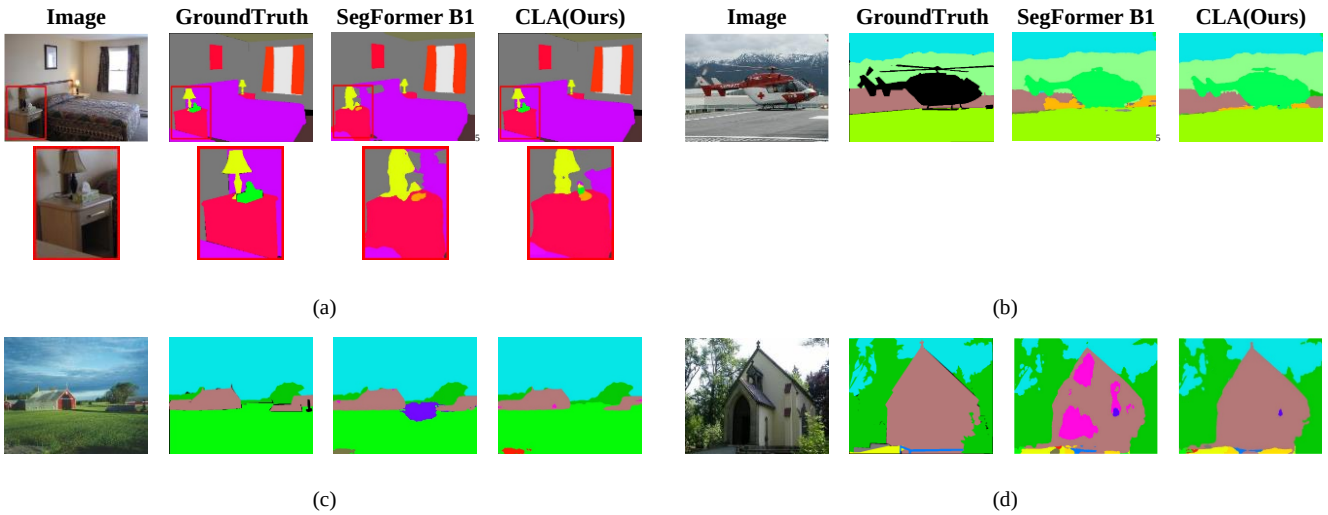

Fig. 3. Visual comparison of non-small objects. Each column from left to right in a subfigure represents the original image, ground truth, SegFormer B1 segmentation results, and our method. (a) lamp (b) helicopter (c) grass (d) church

decoder, ultimately improves the semantic segmentation performance for both small and non-small objects.

It is noteworthy that in this study, only two CLA encoders were used: one takes the feature maps in the first and fourth layer, the other one considers the feature maps in the second and the third layer. However, the CLA mechanism is not limited to these specific layers and can be extended to obtained from any two feature maps. By incorporating information from more layers, this method holds promise for further enhancing the performance of small object semantic segmentation in natural scenes.

This study has successfully integrated SegFormer – B0 and B1 with CLA, which fuses multi-level feature information and compensates for the loss of features during the downsampling process. The proposed design improves the semantic segmentation performance in natural scenes on lightweight models, and the performance on small objects is particularly significant.

Due to the limitation of memory capacity, we were unable to implement SegFormer with AFMA as a baseline comparison to SegFormer with CLA. The implementation of AFMA requires computing cross-feature map attention using the original image (512×512) and the lowest level feature map (16×16). As a result, AFMA requires 128 GB of memory, which is 8 times more than CLA, which restrains us from trying more CLA connection methods. Therefore, in the future, we plan to continuously improve this method and apply CLA to different transformer or CNN architecture models to prove its generalizability on lightweight models.

VI. ACKNOWLEDGMENT

This study is funded by the National Science and Technology Council with grant no. 113-2221-E-033-041- and no. 113-2813-C-033-020-E

REFERENCES

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998, doi: 10.1109/5.726791.

[2] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 21-26 July 2017 2017, pp. 936-944, doi: 10.1109/CVPR.2017.106.

[3] A. Vaswani *et al.*, "Attention is all you need," presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 2017.

[4] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-Aware Trident Networks for Object Detection," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6053-6062, 2019.

[5] B. Singh and L. S. Davis, "An Analysis of Scale Invariance in Object Detection - SNIP," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18-23 June 2018 2018, pp. 3578-3587, doi: 10.1109/CVPR.2018.00377.

[6] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-Object Detection in Remote Sensing Images with End-to-End Edge-Enhanced GAN and Object Detector Network," *Remote Sensing*, vol. 12, no. 9, p. 1432, 2020.

[7] S. Sang, Y. Zhou, M. T. Islam, and L. Xing, "Small-Object Sensitive Segmentation Using Across Feature Map Attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 6289-6306, 2023, doi: 10.1109/TPAMI.2022.3211171.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," Cham, 2015: Springer International Publishing, in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pp. 234-241.

[9] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," arXiv:1706.05587 doi: 10.48550/arXiv.1706.05587.

[10] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: simple and efficient design for semantic segmentation with transformers," presented at the Proceedings of the 35th International Conference on Neural Information Processing Systems, 2024.

[11] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *ArXiv*, vol. abs/2010.11929, 2020.

[12] X. Chu, Z. Tian, B. Zhang, X. Wang, and C. Shen, "Conditional Positional Encodings for Vision Transformers," p. arXiv:2102.10882doi: 10.48550/arXiv.2102.10882.

[13] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21-26 July 2017: IEEE, pp. 633-641, doi: CVPR.2017.544.