HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 1 (2024)

Real Time Semantic Segmentation for Human-Labeled Data: A Comparative Study Between CNN and Transformer

Connie Chen

Cupertino High School, USA

ABSTRACT

Convolutional Neural Networks have rapidly developed in the field of computer vision, notably in image classification, semantic segmentation, and object detection. These networks efficiently extract image features through local receptive fields and shared weights. Despite their effectiveness in various applications, CNNs face limitations, such as challenges in managing large-scale parameters and a tendency to overfit, especially in complex scenarios requiring contextual understanding. On the other hand, Transformer-based models, originally designed for natural language processing, have recently gained prominence in computer vision. They are particularly adept at capturing long-range dependencies, a critical aspect for interpreting complex visual scenes. Their scalability and adaptability open up new avenues for innovation. However, these models also come with drawbacks, including a need for extensive training data and higher computational costs. Their complex structures make optimization particularly challenging in resource-constrained environments. In this paper, our focus is on real-time model comparisons between CNN and Transformer architectures. We represent CNNs with the STDC model and Transformer-based models with the SegFormer. Our analysis revealed that the STDC model significantly outperforms in inference speed, achieving around 97 frames per second (fps), which is notably faster than the 50 fps of the SegFormer B0. However, when it comes to accuracy, the SegFormer B0 demonstrates superiority with a mean Intersection over Union (mIoU) of 86.78, which is more favorable compared to the 82.9 mIoU of the STDC. This study underscores the efficiency and accuracy trade-offs between these two architectures, highlighting the strengths of both CNNs and Transformer-based models in real-time applications.

Introduction

Semantic segmentation, also known as full pixel segmentation and dense prediction, is a fundamental technique used in computer vision to classify pixels of an image into class labels. The Fully Convolutional Network [1], mentioned in the Fully convolutional networks for semantic segmentation, proposed using end-to-end structure with the encoder and decoder structure to approach semantic segmentation. Besides local information, the global context information of images is essential for the semantic segmentation tasks; therefore, PSPNet [2] proposed the use of Pyramid Pooling Module. Later, DeepaLabv3+ [3] used dilated convolution to obtain greater context information. Semantic Segmentation allows further application such as image inpainting, autonomous driving, and object tracking. The global context information enhances the accuracy of the prediction for semantic segmentation. As it generates high resolution outputs, producing per-pixel category prediction, it provides a precise understanding of boundaries and distribution of an image.

CNN significantly increased the performance accuracy of Computer Vision tasks. It has left a notable impact on the application of Semantic Segmentation with fast development. However, CNN has a small effective receptive field, which results in less understanding of global context information. In some cases, classification accuracy would decrease. Fortunately, transformer models have emerged and are now widely used in Computer

Vision. Starting from ViT [4], the self-attention increased accuracy of image classification. Later, the emergence of Swin Transformer [5] and Pyramid vision transformer [6] enabled transformer backbone to undergo downstream tasks, such as semantic segmentation and object detection. Transformers is advantageous in their ability to perform attention across the entire image patch, which results in a larger effective receptive field, compensating for CNN's deficiency.

Real-time refers to the ability for a method to perform a task within a specified time constraint. This ability is critical in cases of segmentation where efficiency is essential for safety reasons. However, a trade-off always exists between model accuracy and efficiency. Different ways are used to address this trade-off. In some cases, the technique of depth-wise separable convolution [7, 8, 9] is applied, which enhances the model's computational speed and efficiency. This innovative approach not only streamlines the model, making it more lightweight, but also maintains a rapid processing capability. This suits real-time application as it increases efficiency. Simplified structures are also used for Transformers to allow faster inference. Encoder-decoder layers might be reduced or employ lightweight attention to meet real-time constraint, such as the transformer referred to in this paper, SegFormer [10]. SegFormer has a hierarchical transformer encoder and lightweight MLP-decoder to increase efficiency.

With the success and popularity of deep learning, semantic segmentation has had significant advancements. There are CNN based and Transformer based semantic segmentation. In an attempt to maximize efficiency and effectiveness, CNN solutions were proposed. CNN is known for its high efficiency, but the downside is its narrow receptive field as a result of the lack of understanding of the global context information of the image. To this end, the Short Term Dense Concatenate (STDC) [11] was proposed. Through multiple continuous layers of encoding images in various scales and receptive fields performance significantly increased. On the other hand, transformer based semantic segmentation is known for its ability to comprehend the global context of an image or scene to enhance the accuracy of the result, but the long and complex attention calculations do not prosper in real time. To overcome this, SegFormer was proposed. Through a positional-encoding-free and hierarchical transformer encoder and lightweight decoder, SegFormer set the new benchmark for transformer based semantic segmentation in terms of efficiency, while maintaining competitive accuracy.

This paper compares the CNN-based and transformer-based semantic segmentation of human photos. We compare STDC and SegFormer in terms of performance, training time, and inference time, as we acknowledge the tradeoff. Humans are often the subject for removal in scenery images; therefore, we focus on human semantic segmentation. Multiple datasets– ADE20K and Cityscapes are combined for better training of the segmentation task for a variety of backgrounds. ADE20K includes data of inner sceneries, while Cityscapes provides data for outer sceneries. Although the PPR10k dataset is not traditionally used for semantic segmentation tasks, the dataset is classified into human and non-human, so we use this dataset to support our human focused semantic segmentation task. In SegFormer structure from B0 to B4, mIoU falls approximately at 86 to 89, which conforms to the expectations of the model structure. For STDC, mIoU falls approximately at 82.9. Thus, we can observe how Transformer based structure excels in inference. On the other hand, CNN structure's inference time can better obtain real-time effect. STDC is 97fps while SegFormer B0-B4 will decrease from 50fps to 15fps.

Related Work

Convolutional Neural Network

Convolutional Neural Network (CNN) processes and analyzes visual data. It is structured with convolutional layers, pooling layers, and fully connected layers. The convolutional layers act as the building block of CNN to extract features of an input image. A kernel (convolution matrix) is used to convolve the image. The pooling layer of CNN reduces spatial dimension by down-sampling feature maps to produce new outputs. This reduces

the amount of computation in the network. The fully connected layers connect information from preceding layers to the output layer through weights matrix and bias vector.

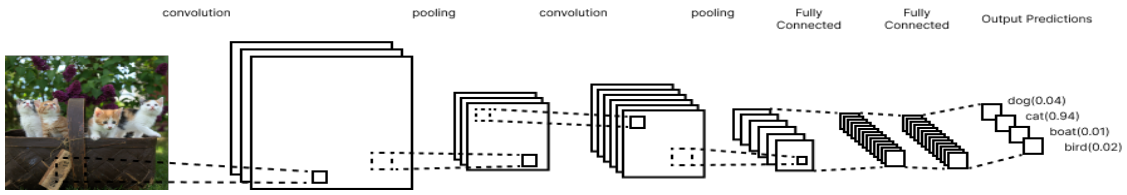

Figure 1. Convolutional Neural Network Architecture

The input image first undergoes convolution, where kernels or filters are applied to different regions to filter information and form a feature map. Pooling is then applied to reduce spatial dimensions of the feature map, which reduces computation time. After that another set of convolutional operations followed by pooling operations is performed to learn additional hierarchical features and preserve important information from the previously extracted information. Fully connected layers classify input into labels by connecting extracted information of previous layers. Finally, the output prediction layer makes predictions based on the learned information.

Transformer

In the traditional structure of CNN, the understanding of local detail is exceeding; however, it is insufficient in understanding global context information. This is due to the small effective receptive field. VIT [4] overcomes this by introducing self-attention application to image classification. ViT had first proved the high-quality performance in image classification. ViT uses multiple transformer layers to make classification. Other methods were introduced to increase the performance level of image classification. PVT [6] and later methods, Swin [5], CvT [12], and CoaT [13] showed the ability for a pure Transformer backbone in dense prediction tasks.

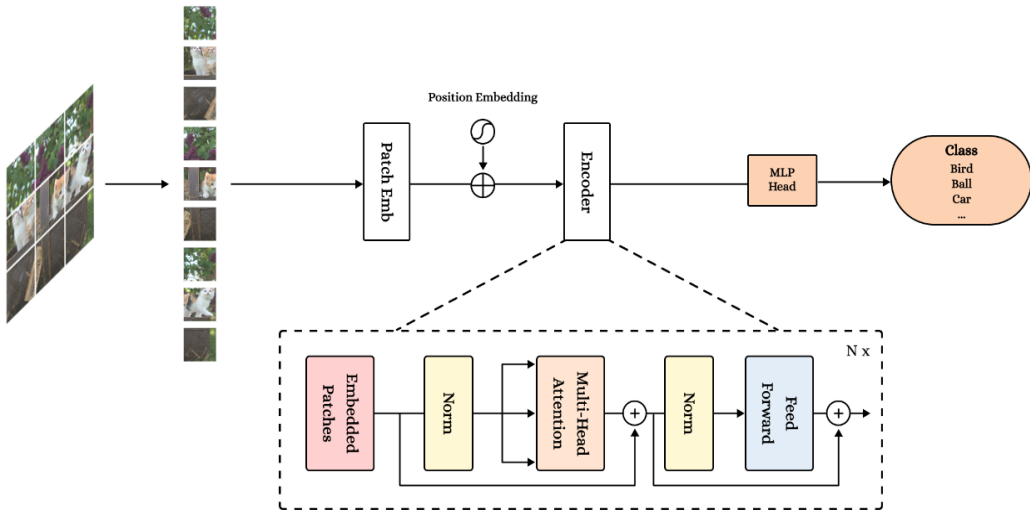

Figure 2. Transformer based Architecture

When an image is given, the image is divided into patches of size 3×3. These patches are used as inputs

and transformed to flattened pixel values. These values along with positional embeddings are fed to the encoder. In the encoder, normalization is applied to the embedding patches, which helps ensure that subsequent layers lie within a certain range. Multi-Head Attention helps understand diverse information across patches. After that, another normalization is applied to maintain smooth learning. Finally, the output undergoes Feed Forward layer to capture complex patterns within patches and process the information.

Real-time Semantic Segmentation

Different ways are used to address the trade-off of real-time semantic segmentation. One way is to limit the input size by resizing, so the complexity of computation would decrease. However, accuracy of the model decreases as prediction around boundaries corrupt due to the loss of spatial details. Another way is to cut feature maps of the network to reduce inference time. The third way is to give up the last stage of the model or down sampling for a tighter framework. Other efficient segmentation methods include lightweight backbone or a multi-branch architecture. DFANet [14] uses a lightweight backbone which increases performance and reduces computational complexity. ICNet [15], with a multi-branch architecture, has a good balance between time and performance.

Method

CNN with Attention for Semantic Segmentation

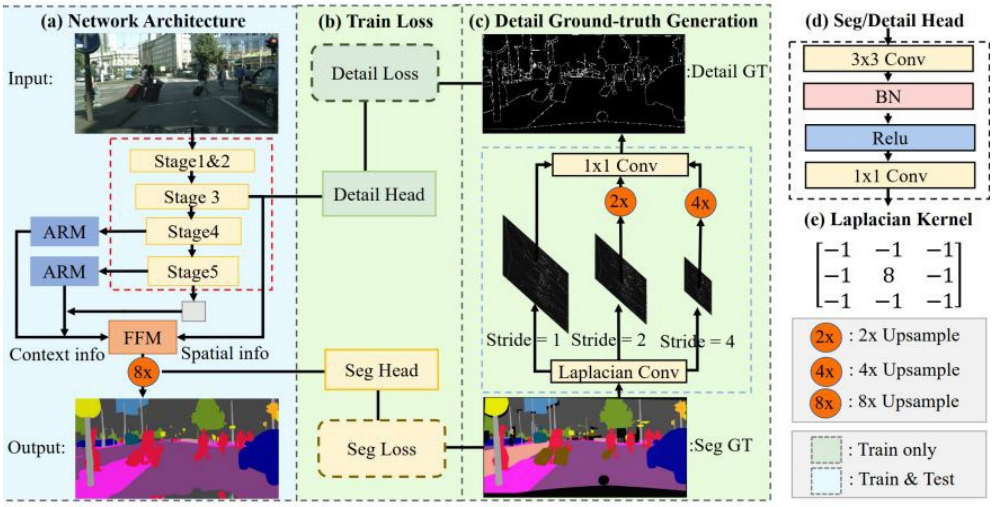

Figure 3. Overview of the STDC Segmentation network

STDC will first undergo multiple stages to extract multi-scale features. When spatial resolution decreases, the internet can extract features from the field of view. This helps understand the image's large structure and background information. Additionally, higher spatial resolution in the initial stage helps extract details and object features. This multi-scale feature extraction allows the model to better understand the global context of the image. The output from stage 4 and 5 will undergo Attention Refine Module (ARM). ARM captures the crucial regions for the task and assigns higher weights to those areas. This refines feature maps and makes it easier to understand global context information with negligible computation cost.

After ARM, the Feature Fusion Module (FFM) combines the low-level features and the high-level

features. The Spatial Path generates low-level features while the Context path generates high-level features. As the two paths generate different levels of feature representation, these features cannot sum up. Therefore, FFM is used to fuse these features by upsampling the low-resolution feature map to correspond to the high-resolution feature map, or vice versa.

Transformer for Semantic Segmentation

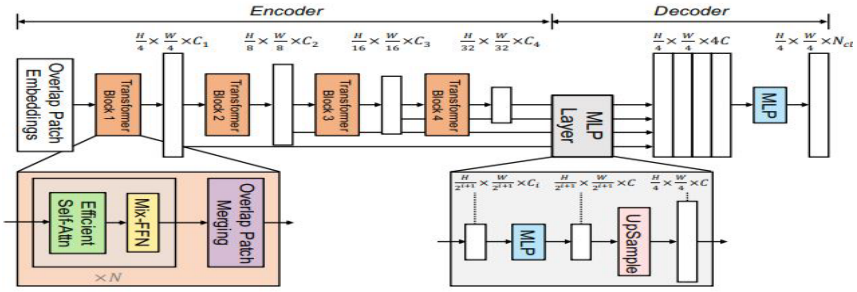

Figure 4. Overview of the SegFormer network
SegFormer has a redesigned encoder-decoder framework different from previous methods. It contains a new positional-encoding-free and hierarchical Transformer encoder and lightweight All-MLP decoder. The new encoder prevents positional codes from interpolating and avoids impacting performance; additionally, it can generate high-resolution and low-resolution features. The lightweight decoder combines local and global attention by collecting information from different layers. Results have shown an improvement in effectives regarding performance and time, while having a robust structure.

Experiments

We compare the results of CNN-based and Transformer-based semantic segmentation with implementation of method on three datasets: ADE20K, Cityscapes, and PPR10k. We introduce the datasets and then explain implementation details. Finally, we discuss the comparison in terms of performance, training time, and inference time.

Dataset and Training Details

Figure 5. The first row contains original images from the datasets PPR10k, ADE20k, Cityscapes, and the second row shows the processed Ground Truth.

ADE20K. ADE20K is a semantic segmentation dataset that covers a variety of visual concepts in scenes. It contains more than 20K scene-centric images with 150 classes of objects. The images are densely annotated with stuff, objects, and object parts labels. It includes scene categories from theLavelMe, SUN and Places database. The dataset consists of classes such as trees, phone, table, streetlight, stove, sky, person etc.

Cityscapes. Cityscapes is a semantic scene labeling dataset of urban street scenes. It contains 5,000 pixel-level annotated images, which is split into 2,975 training set images, 500 validation images, and 1525 test images. 30 classes are included in annotation with 19 classes evaluated for semantic segmentation tasks. Cityscape consists of higher-resolution imagery, making it difficult for real-time semantic segmentation.

PPR10k. PPR10k is a Portrait Photo Retouching (PPR) dataset that contains 11,161 high-quality raw portrait photos (in 1,682 groups). It has both human-region masks for each photo and group-level consistent targets. The dataset covers a broad range of scenes, subjects, and lighting conditions. PPR10k was constructed to facilitate research in automatic PPR tasks.

Implementation details: We convert labels for datasets previously mentioned. This is due to the nature of loss calculation in semantic segmentation, in which 255 labels are neglected. We do this by selecting out images that include humans, labeling the human as 2, and labeling everything else as 1. The following is the ground truth after label conversion of ADE20K, Cityscapes, and PPR10k datasets. We train our SegFormer model for 16,000 iterations for b0 ~ b4, which took approximately 3 to 5 hours. Training STDC for 600,000 iterations took approximately a day, which usually takes five days on a machine with a single NVIDIA 3090 GPU.

Comparison STDC and SegFormer

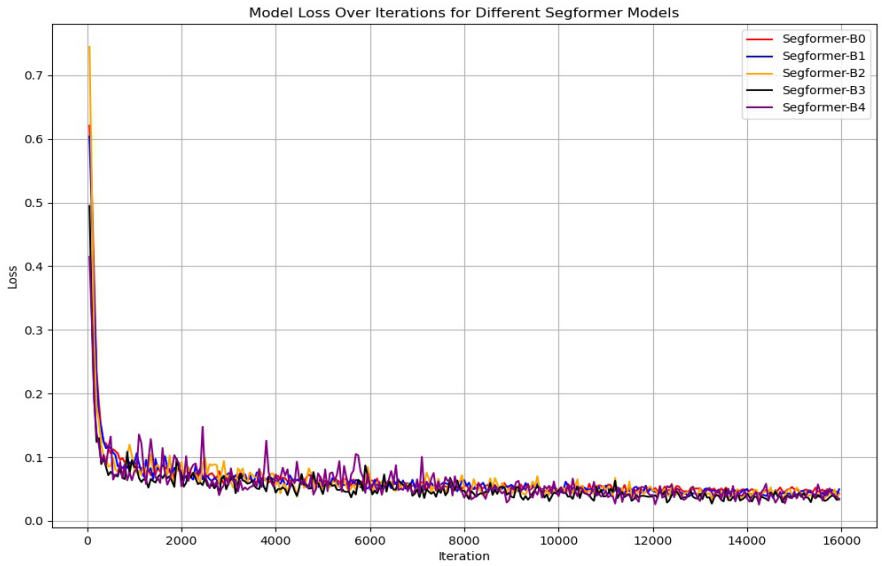

Figure 6. Graph of Training Loss for the SegFormer after 16,000 Iterations.

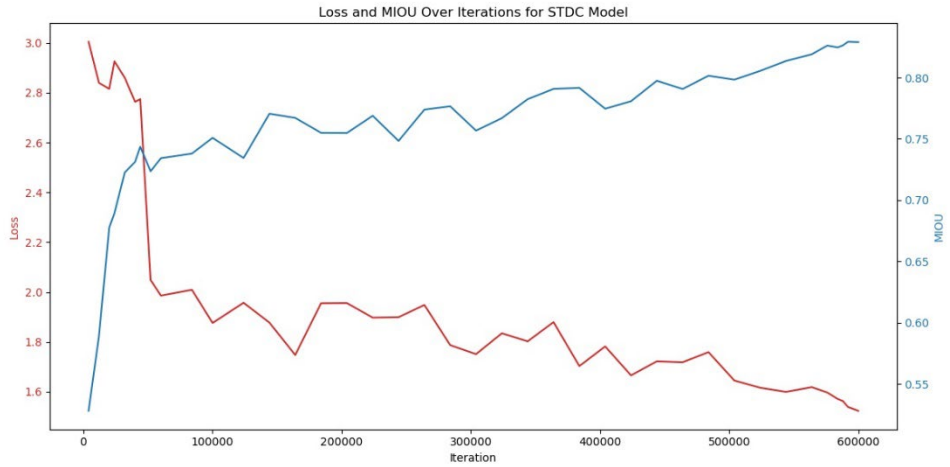

Figure 7. Training Loss and MIOU for the STDC after 600,000 Iterations.

As shown in Fig 4, the loss of the SegFormer presents a straight downward trend before 1000 iterations. Training reaches convergence relatively quickly at 16000 iterations. We attribute this to the reason that Transformer-based model are already pre-trained on a large training set. So, when fine tuned to a smaller dataset, it reaches convergence more quickly and more stably. The chagne in loss after 2000 iterations is fairly stable and remains less than 0.1. We also observe how B0 to B4 SegFormer does not have a big difference in the change in loss, which can be explained by the simplicity of the dataset we use. As the dataset simply consists of human and non-human categories, the results of B0 to B4 SegFormers are mostly the same.

On the other hand, STDC reaches potential convergence after 60,000 iterations. There is even a continuous downward trend in loss and an upward trend in mIoU. It can be observed from Figure 5 that the loss exhibits larger oscillations. This is because the strength of pre-train models for STDC is relatively weak. Therefore, it takes more time to train. However, it is important to note that there is no relationship between model training time and inference time. In fact, inference time for STDC is less than SegFormer.

Overall, SegFormer has higher performance with less training time.
Not finished training

Table 1. MIOU and Inference Time FPS for SegFormer and STDC.

Model	MIOU	Inference Time
SegFormer B0	0.8678	50fps
SegFormer B1	0.8773	32fps
SegFormer B2	0.8858	24fps
SegFormer B3	0.8913	17fps
SegFormer B4	0.8926	15fps
STDC	0.8292	97fps

As expected, transformer-based model achieves better results as shown through the higher mIoU compared to STDC. Moreover, SegFormer reaches convergence at a faster rate. Despite STDC having lower performance and requiring a longer training time, it excels in real-time tasks. STDC inference time is 97fps, while SegFormer's ranges from 15fps to 50fps. STDC's inference speed is faster than that of SegFormer's for more than 50%. For this reason, smartphones generally support CNN operator for commercial use.

Figure 8. Qualitative results on Cityscapes, ADE20K, and PPR10K, from left to right: Input Data, SegFormer B0, SegFormer B4, STDC.

For the first set of images, SegFormer generated a relatively more accurate result. While the STDC

generated image is decent, parts of the hand, arm, and body are cut off, especially in regions near the boundaries. However, for the second set of images, STDC has a higher performance. This may be partly due to the clothing of the person to the right, as the color is similar to the background. So SegFormer had cut it off.

Conclusion

Transformer-based and CNN-based model are compared in this paper to observe results of the human focused semantic segmentation task real-time. SegFormer and STDC are compared in terms of their performance level, training time, and inference time. Results support that SegFormer has a higher level of performance and requires less training time compared to STDC, which undergoes more iterations to reach convergence. STDC, however, requires less inference time. Having these observations, future work can focus on human focused real-time image inpainting tasks.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

Reference

[1] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[2] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2881-2890).
[3] Chen, Liang-Chieh, et al. "Encoder-decoder with atrous separable convolution for semantic image segmentation." Proceedings of the European conference on computer vision (ECCV). 2018.

[4] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[5] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012-10022).

[6] Wang, W., Xie, E., Li, X., Fan, D. P., Song, K., Liang, D., ... & Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 568-578).

[7] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[8] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

[9] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1314-1324).

[10] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, *34*, 12077-12090.

[11] Fan, M., Lai, S., Huang, J., Wei, X., Chai, Z., Luo, J., & Wei, X. (2021). Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9716-9725).

[12] Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., & Zhang, L. (2021). Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 22-31).

[13] Xu, W., Xu, Y., Chang, T., & Tu, Z. (2021). Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9981-9990).

[14] Li, H., Xiong, P., Fan, H., & Sun, J. (2019). Dfanet: Deep feature aggregation for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9522-9531).

[15] Zhao, H., Qi, X., Shen, X., Shi, J., & Jia, J. (2018). Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 405-420).