HAFormer: Unleashing the Power of Hierarchy-Aware Features for Lightweight Semantic Segmentation

Guoan Xu, Wenjing Jia, Tao Wu, Ligeng Chen, Guangwei Gao, *Senior Member, IEEE*

***Abstract*—Both Convolutional Neural Networks (CNNs) and Transformers have shown great success in semantic segmentation tasks. Efforts have been made to integrate CNNs with Transformer models to capture both local and global context interactions. However, there is still room for enhancement, particularly when considering constraints on computational resources. In this paper, we introduce HAFormer, a model that combines the hierarchical features extraction ability of CNNs with the global dependency modeling capability of Transformers to tackle lightweight semantic segmentation challenges. Specifically, we design a Hierarchy-Aware Pixel-Excitation (HAPE) module for adaptive multi-scale local feature extraction. During the global perception modeling, we devise an Efficient Transformer (ET) module streamlining the quadratic calculations associated with traditional Transformers. Moreover, a correlation-weighted Fusion (cwF) module selectively merges diverse feature representations, significantly enhancing predictive accuracy. HAFormer achieves high performance with minimal computational overhead and compact model size, achieving 74.2% mIoU on Cityscapes and 71.1% mIoU on CamVid test datasets, with frame rates of 105FPS and 118FPS on a single 2080Ti GPU. The source codes are available at *https://github.com/XU-GITHUB-curry/HAFormer*.**

***Index Terms*—Semantic segmentation, lightweight, multi-scale feature extraction, local and global context.**

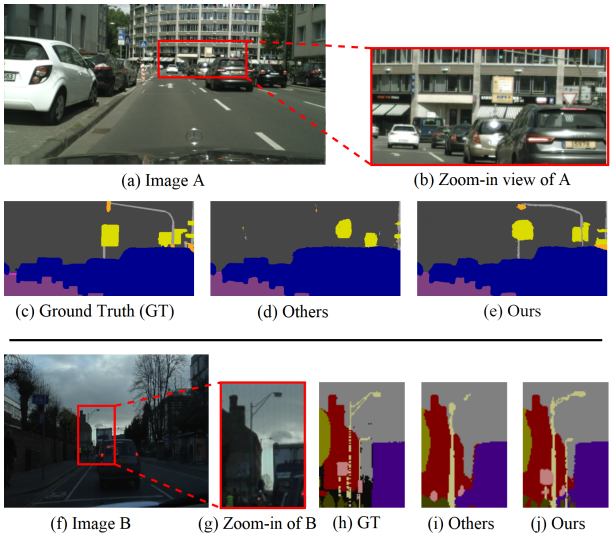

Fig. 1: Visual comparison of small object segmentation using our approach versus an existing method on sample images from Cityscapes (top) and CamVid (bottom).

I. Introduction

SEMANTIC segmentation involves the task of assigning a label to each pixel in a given image, making it a fundamental dense prediction task in computer vision with applications in autonomous driving [1], medical care [2], satellite remote sensing [3], and more. Previous methods, such as [4], [5], leverage deep convolutional neural networks (CNNs) for feature extraction, incorporating feature pyramid structures for multi-scale information perception [6] and attention modules for global context perception [7]–[9]. Although these methods have achieved considerable accuracy, they often require extensive computational resources and exhibit relatively slow inference speeds due to deep network stacking for larger receptive fields and higher semantic levels.

To accommodate devices with limited computational resources, recent studies [10]–[14] have focused on developing lightweight segmentation models. For example, ERFNet [11] employs 1-D non-bottleneck modules to reduce computation, while ICNet [13] utilizes inputs of varying resolutions to enhance information flow across different branches. FBSNet [15] uses a symmetrical encoder-decoder structure with a spatial detail branch and a semantic information branch to refine contextual details. Typically, these models simplify the base module structures to minimize computational costs. However, while enhancing computational efficiency, their segmentation accuracy is often compromised due to the local limitations of convolution networks and shallower network depths.

Transformers have recently demonstrated remarkable success in various computer vision communities [16], [17]. Drawing inspiration from this progress, researchers have started integrating ViT [18] architectures to tackle semantic segmentation challenges. Unlike CNNs, Transformers inherently provide a broad global receptive field through their extensive global attention mechanisms. Models using Transformers as

This work was supported in part by the foundation of Key Laboratory of Artificial Intelligence of Ministry of Education under Grant AI202404, in part by the Open Fund Project of Provincial Key Laboratory for Computer Information Processing Technology (Soochow University) under Grant KJS2274. *(Corresponding author: Wenjing Jia; Guangwei Gao.)*

Guoan Xu and Wenjing Jia are with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia (e-mail: xga_njupt@163.com, Wenjing.Jia@uts.edu.au).

Tao wu and Ligeng Chen are with the State Key Laboratory of Novel Software Technology, Nanjing University. Ligeng Chen is now a researcher and software engineer in Honor Device Co., Ltd.(e-mail: wt@smail.nju.edu.cn, chenlg@smail.nju.edu.cn).

Guangwei Gao is with the Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing, China, and also with the Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai, China (e-mail: csggao@gmail.com).

image encoders excel in global context modeling, leading to significant improvements in segmentation accuracy compared to CNN-based approaches. While UNETR [19] and other methods [20], [21] base predictions on the last layer of the Transformer encoder, they tend to overlook smaller-scale objects in images, affecting the precise classification of smaller elements or pixels, as depicted in Fig. 1. SegFormer [22] introduces a hierarchical attention model integrating a hierarchical Transformer encoder and a lightweight multi-layer perceptron (MLP) decoder to enhance segmentation precision. MPViT [23] effectively incorporates multi-scale feature inputs into Transformer operations, yielding impressive results.

These methods prioritize high segmentation accuracy but often overlook model efficiency. Firstly, transformer-based approaches lack inductive bias, making their training slow and challenging to converge. In addition, they typically require larger datasets and extended training duration, resulting in significant training overhead. Secondly, slow inference speeds are attributed to the time-consuming multi-head self-attention (MHSA) operations. The computational burden escalates, especially with high-resolution inputs, due to the quadratic complexity of MHSA. Additionally, these methods may struggle with capturing details and small objects due to their limited fine local modeling capabilities.

In this work, our goal is to develop a lightweight semantic segmentation model that leverages both CNNs and Transformers, focusing on minimizing model size and computational requirements. Introducing the "HAFormer" model, we combine the global receptive capabilities of Transformers with the local perception strengths of CNNs to unleash the power of hierarchy-aware features.

The core contributions of this paper are threefold:

- We propose a novel Hierarchy-Aware Pixel-Excitation (HAPE) module, utilizing hierarchy and content-aware attention mechanisms to reduce the computational load while enabling the extraction of deeper semantic information from pixels under various receptive fields.
- We develop an effective feature fusion mechanism, named correlation-weighted Fusion (cwF), to synergistically integrate the local and global context features learned by CNNs and Transformers, effectively enhancing accuracy.
- We propose an Efficient Transformer to decompose Q, K, and V matrices, which effectively addresses the quadratic computational complexity challenge present in traditional Transformer models.

Extensive experiments conducted on two widely used benchmarks demonstrate that our HAFormer achieves a balance between segmentation accuracy and efficiency.

The remainder of this paper is structured as follows: Section II provides a comprehensive review of related works. Section III presents the details of our proposed HAFormer, focusing on its three key components. Section IV describes the detailed experimental setting and presents the evaluation results, including ablation studies and discussions. Finally, Section V concludes the paper by summarizing the key findings and discussing future directions.

II. Related Work

A. Hierarchical Methods in Semantic Segmentation

In dense prediction tasks, accurately classifying multi-scale and small target objects is a common challenge. This is particularly evident in semantic segmentation, where the classification of small objects can be affected by neighboring larger objects, leading to misclassification. Hierarchical methods effectively address this challenge by utilizing convolutions with varying dilation rates or pooling layers with different rates. The outcomes are then cascaded or concatenated to integrate information from diverse scales. This multi-scale integration enhances receptive field levels, mitigating ambiguity from varying local region sizes and improving object detail handling. Existing hierarchical approaches [6], [13], [14], [22], [24], [25] can be classified into overall hierarchical structures or specific hierarchical modules, summarized as follows:

Hierarchical Structures. Several approaches have adopted a multi-scale design, featuring distinct network branches handling inputs or feature maps of varying resolutions. A notable method following this approach is ICNet [13], which incorporates three encoding branches (low-resolution, medium-resolution, and high-resolution), each excelling at extracting fine-grained information at different scales to enhance boundary information in the output. In contrast, HRFormer [24] effectively combines robust semantic information with precise location details. Whereas HSSN [26] is a hierarchical approach, focusing on categorizing objects like "Human-Rider-Bicyclist" rather than addressing pixel-level classification challenges for small objects. Other methods, including [14], [22], [25], [27], utilize multi-scale structures by parallelizing multiple resolution branches and facilitating continuous information interaction among them.

Hierarchical Modules. Numerous methods integrate hierarchical modules at specific layers within the architecture, allowing the utilization of varied receptive fields on feature maps. For example, the ASPP module used in DeepLab [28], [29] and DenseASPP [6] effectively extracts features from different scales through atrous convolutions, addressing the variability in object scales within and across images. PSPNet [30] stands out for its pyramid pooling module that integrates features from four scales. By collecting and merging contextual information from diverse scales, this module generates more representative and discriminative features than those from global pooling alone. Models using this module can enhance their recognition capability for objects of various sizes. Inspired by the "wider" modules [30], [31], in this work we demonstrate that utilizing multiple diverse convolution kernels efficiently enhances expressive capacity, leading to improved performance with minimal computational and parameter overhead.

B. Vision Transformer in Semantic Segmentation

The groundbreaking ViT [18] introduces a pure transformer framework for image recognition, treating images as sequences of patches processed through multiple layers. Subsequent models such as DeiT [32], Fact [33], CrossFormer [34], and DViT [35] have further excelled in image processing

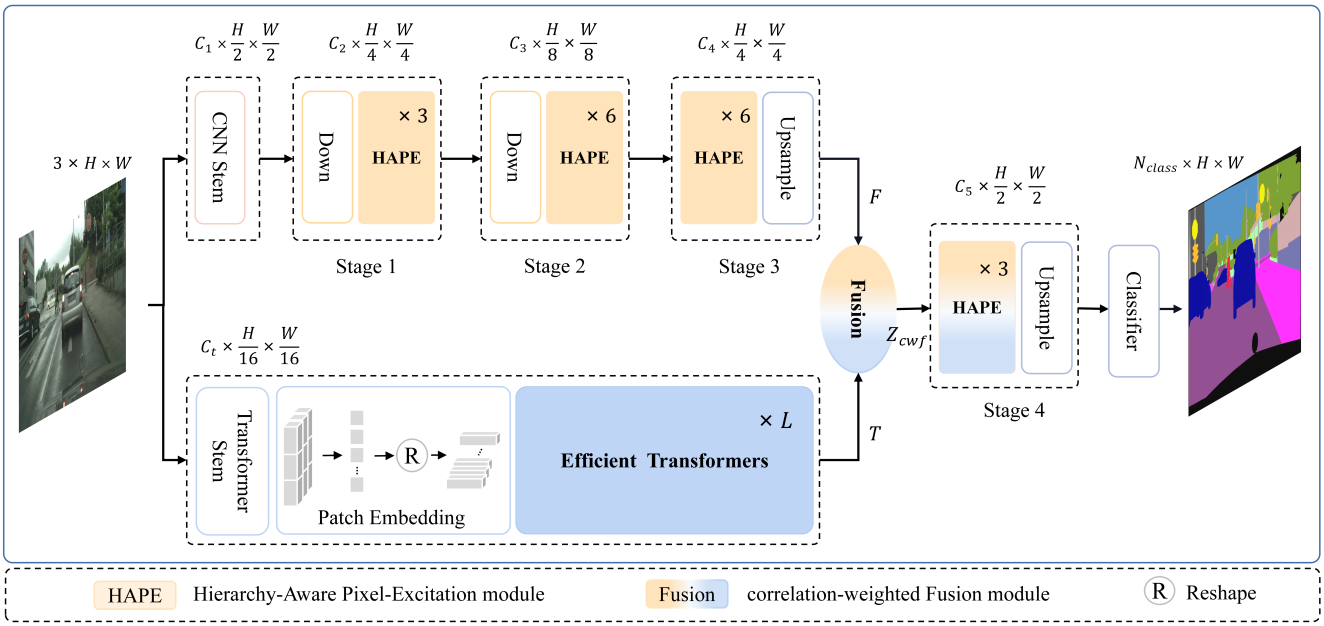

Fig. 2: The overall architecture of the proposed HAFormer. HAFormer introduces a Hierarchy-Aware Pixel-Excitation (HAPE) module for adaptive multi-scale local feature extraction. For global perception modeling, HAFormer develops an efficient Transformer module to streamline the quadratic calculations. Additionally, a correlation-weighted Fusion (cwF) module selectively combines diverse feature representations, markedly boosting predictive accuracy.

tasks. SETR [16] is a tailored paradigm for segmentation, utilizing a pure Transformer model in the encoder and various CNN decoder combinations to achieve state-of-the-art results. Swin-Transformer [17] addresses redundant computations, easing computational loads to some extent. However, these methods still require extensive training data to match CNN performance, posing challenges in dense prediction fields requiring detailed annotations. Transformer-based models such as [23], [27] have recognized the importance of hierarchical perceptions in dense prediction tasks, incorporating multi-scale structures and pyramid modules in their designs.

Recent studies have noted that Transformers often prioritize global long-range dependencies, potentially overlooking critical features like local connections and translation invariance characteristic of CNNs. Consequently, various methods [2], [36]–[38] have sought to combine CNNs and Transformers to fully leverage the strengths of both. However, these efforts struggle to balance real-time inference requirements and low-latency capabilities. Lightweight techniques such as LETNet [39] position the Transformer as a capsule network while others, such as TopFormer [40], integrate it as an auxiliary component in the decoder to enhance boundary recovery. Nonetheless, a definitive solution for effectively combining global and local information remains elusive.

To tackle the challenges of high computational requirements and effectively integrating local information with a global context when combining CNNs with Transformers, our HAFormer introduces an Efficient Transformer (ET) module to manage computational complexity and a correlation-weighted Fusion (cwF) mechanism to harmonize features from CNNs and Transformers.

C. Attention Mechanisms in Semantic Segmentation

Inspired by the focal nature of human visual perception, attention mechanisms emphasize significant features while disregarding irrelevant ones. These mechanisms fall into two main categories: channel attention and spatial attention. In channel attention methods, SKNet [41] enables neurons to dynamically adjust their receptive field sizes based on input scales. Spatial attention methods, such as non-local neural networks [8], capture long-range dependencies in semantic segmentation. However, modeling relationships between all locations can be computationally intensive. Asymmetric non-local neural networks [9] attempted to reduce computational costs, yet they may still be resource-intensive, especially with high-resolution input features.

Researchers have explored combining both channel and spatial attention mechanisms to enhance features from multiple perspectives. For instance, CBAM [7] sequentially operates along two independent dimensions (channel and spatial), producing attention maps that are then multiplied with input features for adaptive feature optimization. DANet [4] and CCNet [42] integrate channel and spatial attention in parallel, employing self-attention operations and combining the resulting features. CAA [43] disassembles axial attention and integrated channel attention to manage conflicts and prioritize features. These methods, utilizing self-attention mechanisms, have demonstrated positive results.

A prevalent challenge involves pixel-wise long-distance modeling, which incurs high computational costs, rendering it unsuitable for deployment in resource-constrained scenarios. This study introduces a lightweight model that optimizes the local perception of CNNs and the global modeling abilities of Transformers. We address the computational complexity

issue by utilizing a spatial reduction-linear projection and split operation strategy within our proposed Efficient Transformer (ET) module.

III. The Proposed Method

A. Overall Architecture

The overall architecture of our HAFormer is illustrated in Fig. 2, which features three components: a CNN encoder enhanced with hierarchy-aware pixel excitation, an efficient Transformer encoder, and a lightweight decoder.

For a given input image $I \in \mathbb{R}^{3\times H\times W}$ with dimensions of $H \times W$, the model begins with a CNN Encoder, producing features $F \in \mathbb{R}^{C_f\times H_f\times W_f}\left(H_f=\frac{H}{8}, W_f=\frac{W}{8}\right)$. Simultaneously, the input I undergoes processing in the Transformer encoder post the Transformer Stem block, resulting in feature embedding $T \in \mathbb{R}^{N\times D}$, where $N=\frac{H_t}{P}\times\frac{W_t}{P}\left(H_t=\frac{H}{16}, W_t=\frac{W}{16}\right)$ denotes the token count, $D=C_t\times P^2$ denotes the dimension of each token, and P signifies token size. Subsequently, the two distinct types of context features, F and T, are synergized effectively by our newly designed correlation-weighted Fusion (cwF) module. This fusion of correlated CNN and Transformer features enhances boundary information and restoration with the lightweight decoder segmentation head.

Specifically, to optimize the CNN encoder, we employ three 3×3 convolutional layers in the CNN Stem block. In this configuration, the last layer has a stride of 2, resulting in a feature map size of $C_1\times\frac{H}{2}\times\frac{W}{2}$, where C_1 denotes the output channel count. In contrast, the Transformer stem in the Transformer Encoder reduces the resolution while extracting feature representations, contributing to the model's lightweight design by minimizing computational load, since higher resolution means more computation. Therefore, in the Transformer Stem block, we employ four 3×3 convolution layers with a stride of 2, resulting in an output feature size of $C_t\times\frac{H}{16}\times\frac{W}{16}$.

B. Hierarchy-Aware Pixel-Excitation (HAPE) Module

Employing convolutions with diverse kernel sizes within the same layer, combined with pixel excitation, facilitates feature extraction from objects of varying sizes. Building on this concept and drawing inspiration from works like [30], [44], we adopt a multi-scale strategy to capture unique pixel features across different receptive field levels. Unlike the layer-wise merging seen in ESPNet [12] and concatenation in Inception [45], our module avoids redundant computations, leading to a more streamlined network while preserving feature effectiveness. Additionally, to further improve pixel representation across diverse scales, we introduce the innovative Hierarchy-Aware Pixel Excitation (HAPE) module in this study. This module enhances the model's ability to effectively recognize objects of various sizes in an image, ultimately reducing pixel misclassification rates.

Specifically, as depicted in Fig. 3, given a feature input $X_{in}\in\mathbb{R}^{N_c\times H_c\times W_c}$, we initially feed it into a 1×1 convolutional layer to reduce its channel dimensions to $\frac{N_c}{4}$, *i.e.*, the output feature map $\widetilde{X}$ is denoted as

$$\widetilde{X}=f_{1\times1}\left(X_{in}\right), \widetilde{X}\in\mathbb{R}^{\frac{N_c}{4}\times H_c\times W_c}. \tag{1}$$

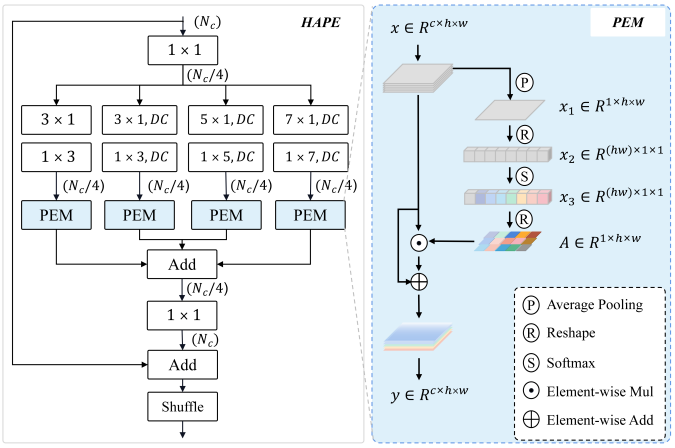

Fig. 3: The architecture of our Hierarchy-Aware Pixel-Excitation (HAPE). DC stands for dilation convolution.

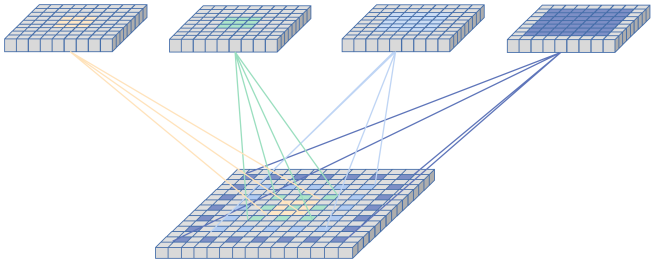

Fig. 4: Supplementary instructions of hierarchical respective fields for HAPE.

Here, $f_{1\times1}$ denotes a convolution operation with a kernel size of 1×1. This dimension reduction facilitates the channel operation in the subsequent hierarchical convolutional layers.

Subsequently, we perform four parallel convolution operations, comprising factorized convolution and depthwise separable convolution, with kernel sizes of 3, 3, 5, and 7, respectively. Additionally, the last three convolutional layers utilize dilated convolution to enhance receptive fields, as shown in Fig. 4. This strategy enables the model to capture image features across various scales, ensuring comprehensive and detailed information extraction.

The above process is expressed as

$$l_1=f_{1\times3}\left(f_{3\times1}\left(\widetilde{X}\right)\right), \tag{2}$$

$$l_i=f^{d_c}_{1\times k_i}\left(f^{d_c}_{k_i\times1}\left(\widetilde{X}\right)\right), \{k_i=3,5,7; i=2,3,4\}, \tag{3}$$

where l_i represents the intermediate features, $f_{1\times k_i}$ is a 1-D convolution operation with a kernel size of k_i, and d_c denotes the dilation rate. For simplicity, some activation and batch normalization operations are excluded from the equations.

A critical element lies in the Pixel-Excitation Module (PEM), which is responsible for enhancing the feature representability through a content-aware spatial-attention mechanism. As illustrated in Fig. 3, the process begins by feeding the input $x\in\mathbb{R}^{c\times h\times w}$ into the Global Average Pooling (GAP) layer, generating $x_1\in\mathbb{R}^{1\times h\times w}$. Subsequently, x_1

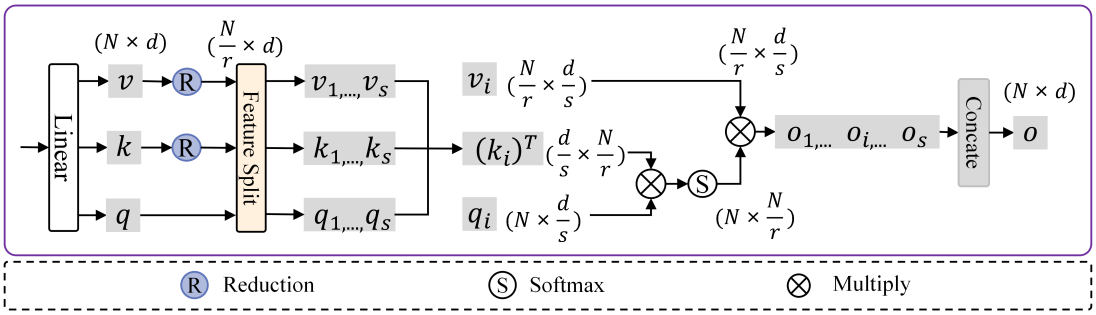

Fig. 5: The architecture of the proposed efficient Multi-Head Self-Attention (eMHSA).

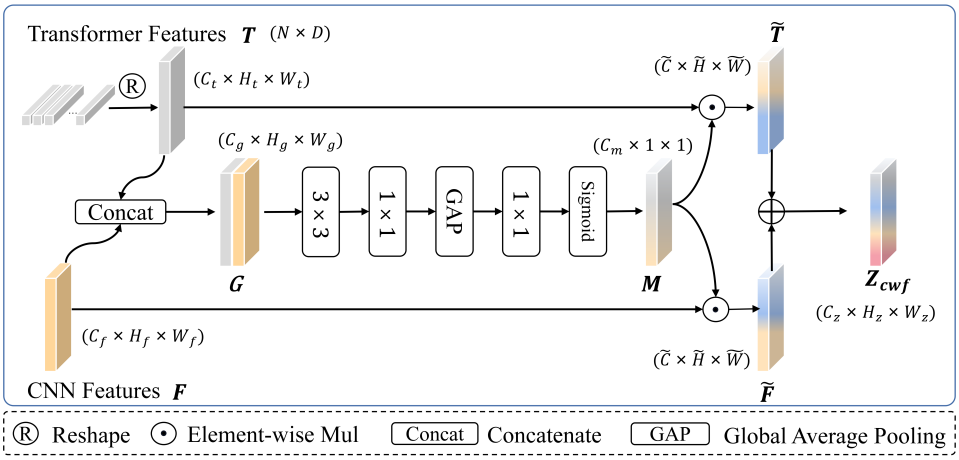

Fig. 6: The architecture of the correlation-weighted Fusion (cwF) module.

undergoes reshaping and flattening operations before being input to the $Softmax$ function to calculate the weight matrix $A \in \mathbb{R}^{1\times h\times w}$. This weight matrix is then multiplied with the input features, resulting in a content-aware attention-enhanced output x'.

This process can be represented as

$$x_1 = \text{Reshape}\left(\text{GAP}\left(x\right)\right), \tag{4}$$

$$A = \text{Reshape}^{-1}\left(\text{Softmax}\left(x_1\right)\right), \tag{5}$$

and

$$x' = \delta\left(x \odot A + x\right). \tag{6}$$

Here, $Reshape$ and $Reshape^{-1}$ denote the reshaping operation and its reverse operation, δ is an activation function, and $\odot$ denotes element-wise multiplication.

Finally, a residual structure is employed to retain the original features, yielding the final output $Y \in \mathbb{R}^{N_c\times H_c\times W_c}$. The four convolution layers are jointly added into a 1×1 convolution for feature fusion and channel restoration. A residual connection is maintained within the module, and the channel shuffle operation effectively facilitates the information interaction between channels, as expressed as

$$Y = \text{Shuffle}\left(f_{1\times1}\left(\delta\left(\sum_{i=1}^{4}\text{PEM}\left(l_i\right)\right)\right) + X_{in}\right), \tag{7}$$

where $Shuffle$ represents the channel shuffle operation, and δ is an activation function.

C. Efficient Transformer

Conventional Transformer methods, as evidenced by [17], [46], can be excessively large for lightweight and real-time models, especially when handling high-resolution inputs. This underscores the urgent need for more efficient Transformers. Inspired by [31], [47], our approach focuses on reducing computational costs by diminishing feature dimensions without significant loss of image details. To achieve this, we introduce a spatial reduction linear projection method which initially maps features into a latent embedding space with reduced dimensions before employing them for multi-head self-attention calculations. This approach, known as efficient Multi-Head Self-Attention (eMHSA) with learned projection and split operation, is depicted in Fig. 5.

Denote the input feature as $X_t \in \mathbb{R}^{C_t\times H_t\times W_t}$, where C_t, H_t, and W_t represent the number of the channel, height, and width of the feature map, respectively. Following the $Reshape$ operation, a sequence of flattened non-overlapping patches is derived, resulting in $X_t \in \mathbb{R}^{N\times\left(C_t\cdot P^2\right)}$, where $N = \frac{H_t W_t}{P^2}$ indicates the number of patches (*i.e.*, the input sequence length), with each patch size being $P \times P$. Subsequently, the patches are mapped by a learnable linear projection layer $E \in \mathbb{R}^{\left(P^2\cdot C_t\right)\times D}$ into a latent D dimensional embedding space, denoted as $Z \in \mathbb{R}^{N\times D}$. This process can be formulated as

$$Z = \left[x_p^1 E; x_p^2 E; ...; x_p^N E\right], \tag{8}$$

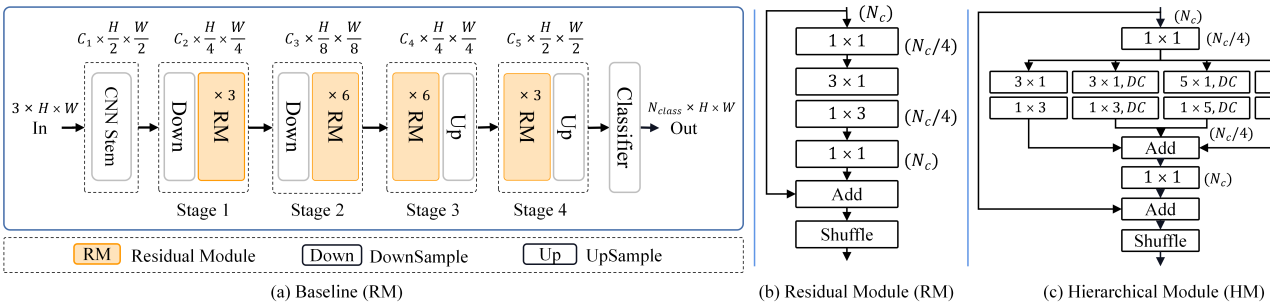

Fig. 7: The architecture of the baseline model (a), the detailed illustration of the Residual Module (RM) used in the baseline (b), and the Hierarchical Module (HM) in our proposed HAPE module (c).

where x_p^i denotes the i-th patch. Note that the omission of position embedding is intentional to allow greater adaptability for different input sizes.

Subsequently, the three matrices in Transformers, namely the queries Q, the keys K, and the values V are derived through their linear projections W^Q, W^K, and $W^V \in \mathbb{R}^{D\times D_h}$. This can be expressed as

$$Q, K, V = ZW^Q, ZW^K, ZW^V \in \mathbb{R}^{N\times D_h}. \tag{9}$$

Moreover, the number of heads h in the multi-head self-attention is also a user-defined parameter, ensuring each head's dimension equals $d = \frac{D_h}{h}$. Consequently, the dimensions of q, k, and v in the i-th head are $N \times d$. In the i-th head, k and v undergo spatial reduction by a factor of r, where r is the reduction ratio and set to 2. Then, the sub-tokens resulting from the feature split operation undergo matrix multiplication with a field representing only $\frac{1}{s}$ of the original perception, where s denotes the number of feature splits, which is set to 4. This process can be described as

$$\left(q_{1,\ldots,}q_s\right), \left(k_{1,\ldots,}k_s\right), \left(v_{1,\ldots,}v_s\right) = \text{Feature\_Split}\left(q, k, v\right). \tag{10}$$

Therefore, the spatial distribution becomes $q_i \in \mathbb{R}^{N\times\frac{d}{s}}$, $k_i \in \mathbb{R}^{\frac{N}{r}\times\frac{d}{s}}$, and $v_i \in \mathbb{R}^{\frac{N}{r}\times\frac{d}{s}}$. This idea shares similarities with the concept of group convolution and can efficiently reduce memory consumption. Thus, the self-attention in the n-th head is calculated as

$$o_i\left(q_i, k_i, v_i\right) = \text{Softmax}\left(\frac{q_i(k_i)^T}{\sqrt{d}}\right) v_i, i \in [1, s], \tag{11}$$

and

$$head^n = \text{Concat}[o_1, o_2, ..., o_s], n \in [1, h], \tag{12}$$

where $Concat[.,.]$ denotes the concatenating operation.

Thus, the final output of the eMHSA is denoted as

$$\text{eMHSA} = \text{Concat}[head^1, head^2, ..., head^h]W^O, \tag{13}$$

where h represents the number of heads in eMHSA, while $W^O \in \mathbb{R}^{D_h\times D}$ serves as a linear projection to restore the dimension. Hence, with the structure designed above, we have reduced the complexity from $O\left(N^2\right)$ to $O\left(\frac{N^2}{sr}\right)$.

It is noteworthy that the Transformer series [16], [17], [23] also utilize a kind of self-attention mechanism, including Multi-Head. However, their approach is computationally intensive for capturing detailed relationships among features, which deviates from our objectives.

As for the MLP layer, we follow the approach described in [31], [48], replacing fixed-size position encoding with zero-padding position encoding. Moreover, we introduce a depth-wise convolution with a padding size of 1 to capture local continuity in the input tensor between the fully connected (FC) layer and GELU in the feed-forward networks. By eliminating fixed-size positional embeddings, the model becomes versatile in handling inputs with different resolutions. Thus, the output of the efficient MLP layer, denoted as "$eMLP$", can be written as

$$\text{eMLP} = \rho\left(\xi_{\text{GELU}}\left(f_{\text{DWConv}}\left(\rho\left(x_e\right)\right)\right)\right), \tag{14}$$

where ρ denotes the FC layer operation, ξ_{GELU} represents the GELU activation function, f_{DWConv} signifies depthwise convolution, and x_e is the input of eMLP.

D. Correlation-weighted Fusion

Numerous studies, such as [2], [36], [40], [46], have explored integrating features from both Transformers and CNNs. For example, SegTransConv [36] introduces a hybrid architecture combining Transformers and CNNs in series and parallel, yet it does not fully exploit the collaborative potential of both. Given the distinct characteristics and computational mechanisms of Transformers and CNNs, conventional element-wise addition or concatenation operations may not yield optimal results. A design leveraging the complementary strengths of both is therefore crucial for maximizing the representability of the extracted features and facilitating information recovery during decoding.

In this paper, we introduce an effective strategy to bridge this gap. Our approach seamlessly combines the distinct types of features extracted by Transformers and CNNs through correlation-weighted integration. By fusing CNN and Transformer features with high correlation, we develop a new correlation-weighted Fusion (cwF) module.

As depicted in Fig. 6, T and F denote intermediate features from the Transformer and CNNs, respectively. Initially, the Transformer feature T is reshaped to match the same shape of the CNN feature F, which is followed by the post-concatenation operation of the two feature sets. To reduce the computational costs, depthwise separable convolution is employed for channel dimensional reduction. Subsequent to

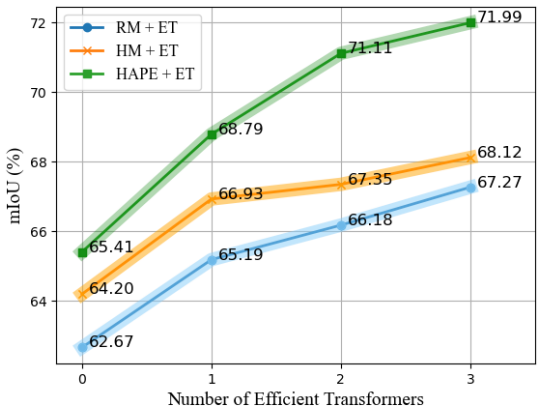

Fig. 8: Performance comparisons using different numbers of ETs with the baseline RM module, HM module, and our HAPE module obtained on the CamVid dataset.

GAP and Sigmoid operations, a correlation coefficient matrix, denoted by M, is calculated. This matrix is then multiplied with the original features to obtain $\widetilde{F}$ and $\widetilde{T}$, which are added together to produce the final output Z.

This process can be expressed as

$$G = \text{Concat}[\text{Reshape}\left(T|F\right), F], \tag{15}$$

where $G \in \mathbb{R}^{C_g \times H_g \times W_g}$ $(C_g = C_f + C_t)$, $Concat$ represents the concatenating operation, and $a|b$ represents the feature map of the size a being restored to size b. Then, the correlation coefficient matrix M can be computed as

$$M = \delta\left(f_{1\times1}\left(\text{GAP}\left(f_{1\times1}\left(f_{3\times3}\left(G\right)\right)\right)\right)\right), \tag{16}$$

where $M \in \mathbb{R}^{C_m \times 1 \times 1}$, δ is the Sigmoid function, GAP represents global average pooling operation, and $f_{k_1\times k_2}$ denotes convolutional operation with a kernel size of $k_1 \times k_2$.

Thus, the resultant cwF features, denoted by Z_{cwF}, can be expressed as

$$Z_{\text{cwF}} = \varphi\left(\widetilde{T} + \widetilde{F}\right), \left\{\widetilde{T} = T \odot M, \widetilde{F} = F \odot M\right\}, \tag{17}$$

where $Z_{\text{cwF}} \in \mathbb{R}^{C_z \times H_z \times W_z}$, φ is the ReLU activation function, and $\odot$ represents element-wise multiplication.

It is noteworthy that feature correlation has also been explored in CTCNet [38], where the correlation between the features derived from Transformers and CNNs is calculated. However, in CTCNet, the module merely concatenates the correlation with the Transformer and CNN features, which cannot effectively align these two types of features, potentially leading to performance degradation due to feature mismatch.

IV. Experiments

To demonstrate the effectiveness of our HAFormer and its individual modules qualitatively and quantitatively, comparative experiments are conducted on benchmark datasets and compared with state-of-the-art (SOTA) approaches. In this section, we first outline the datasets, loss functions, hardware platform configuration, and parameter settings used in our experiments. Then, we present the series of ablation experiments conducted to validate the effectiveness of the individual modules. Finally, comparative experiments are conducted to demonstrate the superiority of our approach over the SOTA approaches.

TABLE I: Ablation studies on the HM and PEM components of the proposed HAPE module. *Cit.*, *Cam.*, and *Param.* denote Cityscapes, CamVid, and Parameter, respectively.

Architecture	Param. (K)↓	FLOPs (G)↓		mIoU (%)↑	
		Cit.	Cam.	Cit.	Cam.
Baseline (RM)	424.912	10.166	5.957	66.78	62.67
Baseline (HM)	450.256	10.401	6.094	$68.25^{1.47\uparrow}$	$64.20^{1.53\uparrow}$
Baseline (HAPE)	482.512	10.402	6.095	$68.91^{2.13\uparrow}$	$65.41^{2.74\uparrow}$
HAFormer (Ours)	602.298	11.051	6.475	$74.18^{7.40\uparrow}$	$71.11^{8.44\uparrow}$

TABLE II: Ablation studies on the impact of Dilation Rates.

Architecture	Dilation Rate				mIoU (%)↑
	Stage1	Stage2	Stage3	Stage4	
Baseline (HAPE)	(1,1,1)	(1,1,1,1,1,1)	(1,1,1,1,1,1)	(1,1,1)	68.91
	(2,2,2)	(2,2,2,2,2,2)	(2,2,2,2,2,2)	(2,2,2)	$69.37^{0.46\uparrow}$
	(2,2,2)	(4,4,8,8,16,16)	(4,4,8,8,16,16)	(2,2,2)	$70.12^{1.21\uparrow}$
HAFormer	(1,1,1)	(1,1,1,1,1,1)	(1,1,1,1,1,1)	(1,1,1)	72.45
	(2,2,2)	(2,2,2,2,2,2)	(2,2,2,2,2,2)	(2,2,2)	$73.16^{0.71\uparrow}$
	(2,2,2)	(4,4,8,8,16,16)	(4,4,8,8,16,16)	(2,2,2)	$74.18^{1.73\uparrow}$
HAFormer (ours)	(2,2,2)	(4,4,8,8,16,16)	(4,4,8,8,16,16)	(2,2,2)	**74.18**

A. Datasets

Our HAFormer model is designed to tackle challenges related to scale variations and contextual information in street scenes. The Cityscapes [49] and CamVid [50] datasets are two prominent benchmarks widely utilized in street scene segmentation research. Hence, to showcase the efficacy of our model, we conducted a series of comprehensive empirical evaluations on these two datasets.

Cityscapes. This dataset comprises 5,000 high-quality images annotated at the pixel level. Captured from various urban settings in 50 cities, these images have a resolution of $2,048 \times 1,024$ and primarily depict driving scenes. The dataset is divided into three subsets: 2,975 images for training, 500 for validation, and 1,525 for testing. While the dataset includes labels for 34 categories, our study focuses specifically on 19 essential semantic categories. We utilize the Cityscapes' built-in tools to adjust the labels to suit our research needs.

CamVid. This is a public dataset of urban road scenes released by the University of Cambridge. The images, with a resolution of 960×720, are captured from a driving perspective, increasing the diversity of observed targets. With over 700 labeled images, the dataset is suitable for supervised learning. The CamVid dataset usually employs 11 common categories for evaluating segmentation accuracy. These categories offer a thorough representation of objects in urban road scenes, making them a valuable resource for research.

B. Implementation Details

The HAFormer model is executed on a single RTX 2080 Ti GPU card with 12GB memory, using CUDA 10.1 and PyTorch

TABLE III: Performance comparison between using TT and ET in HAPE on the Cityscapes (512×1024) and CamVid (360×480) datasets.

Architecture	TT	ET	Param. (K)↓	FLOPs (G)↓		Speed (FPS)↑		mIoU (%)↑	
				Cit.	Cam.	Cit.	Cam.	Cit.	Cam.
HAFormer	✓		760.293	13.341	8.135	56	77	74.66	71.47
		✓	602.298	11.051	6.095	105	118	74.18	71.11

1.8.1. The architecture is trained from scratch without any pre-trained models. We employ Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a weight decay of $1e-5$, along with the "Poly" learning rate policy for optimization.

For Cityscapes, the initial learning rate is $4.5e-2$, and the batch size is set to 5 to maximize GPU memory usage. For CamVid, the initial learning rate is $1e-3$, with a batch size of 8. Following the existing practice, we apply data augmentation techniques including horizontal flipping, random scaling, and random cropping to introduce diversity in the training data, with random scales ranging from 0.25 to 2.0 and the cropping size of 512×1024 for Cityscapes over 1,000 epochs. No post-processing is applied for a fair comparison.

Finally, following the existing practice, the performance is quantitatively evaluated using the averaged mean Intersection-over-Union (mIoU) across all categories, as well as the parameter counts, FLOPs and GPU usage, and processing speed.

C. Ablation Studies

In this part, we conduct a series of ablation experiments to validate the effectiveness of each module in our method.

Ablation Study of the HAPE Module.

In our HAPE module (see Section III-B), we proposed four parallel convolution operations to capture image features across various hierarchies comprehensively. This is then followed by the PEM, designed to enhance the feature representability through a content-aware spatial-attention mechanism. In this section, we show the effectiveness of the hierarchical approach (denoted as "HM") and the PEM approach of our HAPE module, respectively.

The baseline model used for comparison is structured as a single-line type (as shown in Fig. 7), incorporating the standard Residual Modules (RMs). To showcase the performance gains brought by the HM and PEM, we first substitute the RM of the baseline model with the HM module, omitting the PEM part, and then include both the HM and PEM modules to test the effectiveness of the entire HAPE module.

Table I highlights the superior performance of the HM, showcasing mIoU gains of 1.47% and 1.53% over the RM. The HM excels in extracting robust features, facilitating deep semantic information extraction effectively. Moreover, the multi-scale structure significantly enhances the model's performance in feature extraction and small object recognition. Introducing the PEM further enhances segmentation accuracy by 2.13% and 2.74% on both datasets.

Throughout this experiment, the dilated convolution rates are set to 1 in both HM and HAPE to ensure a fair comparison.

TABLE IV: Ablation studies of the ET, the number of ETs in HAPE ("L"), and the cwF module on the Cityscapes dataset.

Architecture	**Fusion**			**Param. (K)↓**	**FLOPs (G)↓**	**mIoU (%)↑**
	Add	Concat	**cwF**			
Baseline (HAPE)	–	–	–	482.512	10.402	70.12
HAFormer (L=1)	✓			549.364	10.780	$71.23^{1.11\uparrow}$
		✓		585.556	12.648	$71.66^{1.54\uparrow}$
			✓	554.742	10.952	$72.50^{2.38\uparrow}$
HAFormer (L=2)	✓			596.920	10.879	$72.28^{2.16\uparrow}$
		✓		633.112	12.747	$73.17^{3.05\uparrow}$
			✓	602.298	11.051	$74.18^{4.06\uparrow}$
HAFormer (ours)			✓	602.298	11.051	**74.18**

Fig. 8 also verifies the efficacy of our HAPE module when being integrated with the Transformer module.

Ablation Study of the Dilation Rates. In this section, we explore how the chosen dilation rates impact segmentation performance. With a consistent number of modules, a larger dilation rate expands the receptive field, allowing the model to perceive a broader scope, and hence is essential for comprehensive feature extraction.

Results shown in Table II reveal that transitioning the dilation rate from all 1s to all 2s (the first two rows) in dilated convolution boosts mIoU by about 0.5%. Further, by progressively increasing the dilated convolution rate in Stages 2 and 3, we observe performance enhancements of 1.21% and 1.73% on the two datasets. Hence, to preserve spatial details, in our approach, we allocate three modules in Stages 1 and 4 while employing six modules in Stages 2 and 3 to capture intricate semantic information within the network's depth. This strategy optimizes calculations for the transformer encoder, improving long-range dependency modeling.

Ablation Study of the Efficient Transformer.

As detailed in Section III-C, another key contribution we made in the HAFormer is the Efficient Transformer (ET) module, which reduces the dimension of features by projecting them into an optimal latent embedding space before calculating self-attention. Table III showcases the performance gains brought by the ET module over the traditional Transformer (denoted as "TT") in terms of segmentation accuracy and computation complexity on Cityscapes and CamVid datasets.

As shown in Table III, the ET design demonstrates a superior balance between efficiency and accuracy. Compared to the traditional Transformer "TT", ET achieves an 18% reduction in parameter count and a 17% decrease in computational load, with only a slight mIoU loss of 0.4%. This results in a more efficient model with minimal impact on performance, and it even offers faster inference speed. In addition, the results in Table IV also reveal the significant enhancement upon integrating features learned through the Transformer, with a remarkable 2.16% boost in mIoU. This underscores the Transformer's exceptional ability to capture long-range dependencies, a feature that the CNN alone cannot achieve.

Additionally, in the proposed HAFormer, the number of ET layers L is deliberately limited to 2, considering computation hardware constraints and also aiming to achieve the best balance under constraints. Although stacking more ET

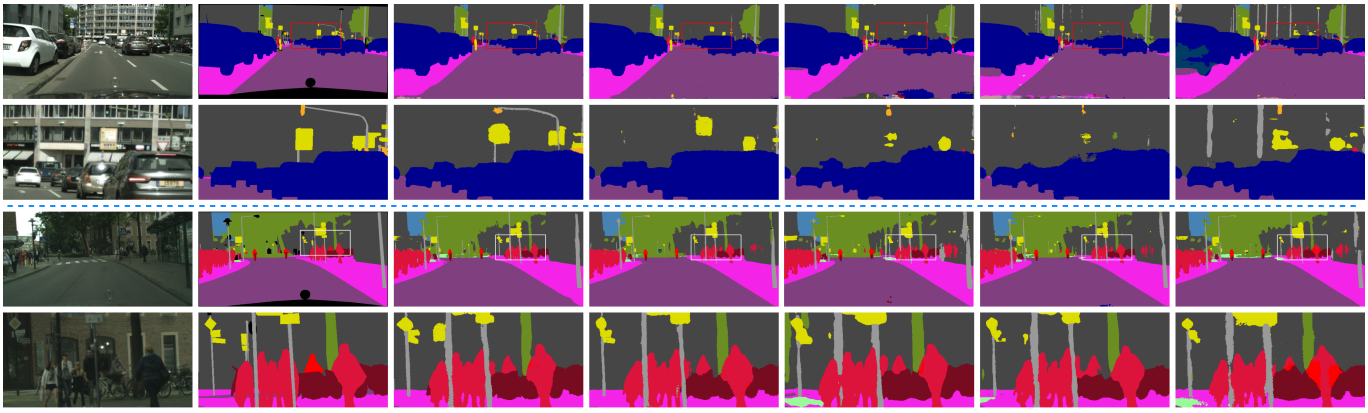

Fig. 9: Visual results on Cityscapes dataset. From left to right: original images, ground truths, predictions of **HAFormer**, LETNet [39], SGCPNet [51], DABNet [52], CGNet [53]. Note that two examples are shown. In each group, the first row visualizes the overall segmentation results, while the second row visualizes the zoom-in of the small areas enclosed in rectangles.

layers could yield better accuracy results, as shown in Fig. 8, the performance gains slow down dramatically when L is greater than 2. Moreover, adding excessive ET layers on a high-resolution dataset like Cityscapes may negatively impact parameters, computations, and inference speed, potentially causing overfitting.

Ablation Study of the Correlation-weighted Fusion. To address the feature mismatch issue between CNNs and Transformers and ensure effective feature restoration during decoding, in Section III-D we introduced the cwF mechanism. Table IV compares the results obtained with our cwF method and two other fusion techniques, *i.e.*, element-wise addition, and concatenation. The table illustrates enhanced segmentation accuracy when integrating CNN and Transformer features using all three fusion methods. Notably, our cwF achieves a performance improvement of 2.38% over the baseline with one ET layer and 4.06% gain with two layers stacked.

Moreover, from Table IV we observe that (a) Compared to the simple element-wise addition fusion scheme, our cwF shows performance gains of 1.27% and 1.90% in the two cases with only a slight increase in parameter count and FLOPs; (b) Our cwF presents mIoU gains of 0.84% and 1.01% over the computationally expensive concatenation operation, respectively, while achieving about 5% reduction in parameter count and 15% decrease in computational load. These experimental outcomes further demonstrate the effectiveness of our cwF.

D. Comparisons with SOTA Methods

In this section, we extensively assess and compare the performance and efficiency of our method against some state-of-the-art approaches to showcase the advantages of our proposed method. Our evaluation centers on three key aspects: segmentation accuracy, model parameters, and floating-point operations (FLOPs).

Evaluation Results on Cityscapes. Quantitative comparisons with advanced semantic segmentation methods on the Cityscapes test set are presented in Table V. Per-class results are detailed in Table VI, and visualization outcomes are displayed in Fig. 9. To ensure fairness, no augmentation techniques are used during testing, and data for other networks are referenced from pertinent sources. Contemporary semantic segmentation models fall into two main categories: those emphasizing larger size and higher precision, and those prioritizing real-time practicality with a balance between accuracy and efficiency.

While larger models achieve high accuracy, their FLOPs and speed lag behind lightweight models, making them unsuitable for real-time processing on devices with limited resources. In contrast, lightweight models like ENet [10], ESPNet [12], CGNet [53], and FPENet [59] are computationally efficient. Despite their reduced parameter count, their overall performance, especially in accuracy, is lacking. In terms of accuracy, EFRNet-16 [66] shows similarities to our results. However, it is noteworthy that its parameter count and GFlops are 2 times greater than ours. Apparently, our model requires fewer parameters and computations, highlighting the efficiency of our approach.

Evaluation Results on CamVid. To further validate the effectiveness and generalization capacity of our model, we compared it with other lightweight methods on the CamVid dataset, as shown in Table VII. While MGSeg [64] excels in accuracy, surpassing our method by 1.6 points, it does so at the cost of having 22 times more parameters than ours, indicating an unfavorable trade-off. On the other hand, SGCPNet [51] exhibits notable speed but lacks accuracy. In contrast, our HAFormer has achieved a better balance between these aspects. The lower overall performance on the CamVid dataset, compared to Cityscapes, is due to its smaller size and lower resolution, which highlights the robust generalization capability of our approach. Visualization results in Fig. 10 further demonstrate the advantages of our HAFormer.

Speed Comparison. To ensure a fair comparison, all methods are executed on the same platform, as the computational load directly impacts inference speed, which can vary depending on the device. In our controlled evaluation, a single NVIDIA RTX 2080Ti GPU is utilized to measure model execution times. The comparison of speed and runtime between our proposed HAFormer and other lightweight meth-

TABLE V: Comparisons on the Cityscapes test dataset. "$*$" indicates that the method utilizes multiple graphics cards. "–" means that the result is not provided in the corresponding methodology.

	Methods	Year	Resolution	Backbone	Param. (M) ↓	FLOPs (G) ↓	GPU	Speed (FPS) ↑	mIoU (%) ↑
Large Model	SegNet [54]	2017 TPAMI	360 × 640	VGG-16	29.50	286.0	*	17.0	57.0
	DeepLab [28]	2017 TPAMI	512 × 1024	ResNet 101	262.10	457.8	*	0.3	63.5
	PSPNet [30]	2017 CVPR	713 × 713	ResNet-101	68.10	2048.9	*	1.2	78.5
	DeepLab-V3+ [29]	2018 ECCV	–	MobileNet-V2	62.70	2032.3	*	1.2	80.9
	DANet [4]	2019 CVPR	1024 × 1024	ResNet-101	66.60	1111.8	*	4.0	81.5
	SETR [16]	2021 CVPR	768 × 768	ViT-Large	318.30	–	*	0.5	82.2
	HRViT [55]	2022 CVPR	1024 × 2048	HRViT	28.60	66.8	*	–	83.2
	SegFormer [22]	2021 NIPS	1024 × 2048	MiT-B5	84.70	1447.6	*	2.5	84.0
	Lawin [56]	2022 arxiv	1024 × 1024	Swin-L	–	1797.0	*	–	**84.4**
	DDPS-SF [57]	2023 arxiv	1024 × 2048	MiT-B5	122.80	–	*	–	82.4
Lightweight Model	ENet [10]	2016 arxiv	512 × 1024	No	**0.36**	3.8	Titan X	135	58.3
	ESPNet [12]	2018 ECCV	512 × 1024	ESPNet	**0.36**	–	Titan XP	113	60.3
	NDNet [58]	2021 TITS	512 × 1024	No	0.50	**3.5**	Titan X	101	61.1
	CGNet [53]	2021 TIP	360 × 640	No	0.50	6.0	V100	120	64.8
	ERFNet [11]	2017 TITS	512 × 1024	No	2.10	–	Titan X	42	68.0
	ICNet [13]	2018 ECCV	1024 × 2048	PSPNet-50	26.50	28.3	–	30	69.5
	DABNet [52]	2019 BMVC	1024 × 2048	No	0.76	42.4	1080 Ti	28	70.1
	FPENet [59]	2019 BMVC	512 × 1024	No	0.40	12.8	Titan V	55	70.1
	LEDNet [60]	2019 ICIP	512 × 1024	No	0.94	–	1080 Ti	71	70.6
	FBSNet [15]	2023 TMM	512 × 1024	No	0.62	9.7	2080 Ti	90	70.9
	SGCPNet [51]	2022 TNNLS	1024 × 2048	MobileNet	0.61	4.5	1080 Ti	103	70.9
	MSCFNet [14]	2022 TITS	512 × 1024	No	1.15	17.1	Titan XP	50	71.9
	SegFormer [22]	2021 NIPS	512 × 1024	MiT-B0	3.80	17.7	V100	48	71.9
	MLFNet [61]	2023 TIV	512 × 1024	ResNet-34	13.03	15.5	Titan XP	72	72.1
	BiseNet-V2 [62]	2021 IJCV	1024 × 2048	Xception	3.40	21.2	1080 Ti	**156**	72.6
	PCNet [63]	2022 TITS	1024 × 2048	Scratch	1.63	11.8	2080 Ti	72	72.7
	MGSeg [64]	2021 TIP	1024 × 1024	ShuffleNet-V2	4.50	16.2	Titan XP	101	72.7
	LETNet [39]	2023 TITS	512 × 1024	No	0.95	13.6	3090	150	72.8
	SegTransConv [36]	2023 TITS	512 × 1024	STDC	7.00	10.2	3090	57	73.0
	PMSDSEN [65]	2023 ACM MM	512 × 1024	No	0.92	10.2	–	53	73.2
	EFRNet-16 [66]	2022 TMM	512 × 1024	EAA	1.44	25.1	Titan X	58	**74.3**
	HAFormer (Ours)	-	512 × 1024	No	0.60	11.1	2080 Ti	105	**74.2**

TABLE VI: Comparisons with other methods about per-class results on the Cityscapes test set. Roa: Road, Sid: Sidewalk, Bui: Building, Wal: Wall, Fen: Fence, Pol: Pole, TLi: Traffic Light, TSi: Traffic Sign, Veg: Vegtation, Ter: Terrain, Sky: Sky, Ped: Pedestrain, Rid: Rider, Car: Car, Tru: Truck, Mot: Motorcycle, Bic: Bicycle.

Methods	Roa	Sid	Bui	Wal	Fen	Pol	TLi	TSi	Veg	Ter	Sky	Ped	Rid	Car	Tru	Bus	Tra	Mot	Bic	mIoU
SegNet [54]	96.4	73.2	84.0	28.4	29.0	35.1	39.8	45.1	87.0	63.8	91.8	62.8	42.8	89.3	38.1	43.1	44.1	35.8	51.9	57.0
ENet [10]	96.3	74.2	75.0	32.2	33.2	43.4	34.1	44.0	88.6	61.4	90.6	65.5	38.4	90.6	36.9	50.5	48.1	38.8	55.4	58.3
ESPNet [12]	97.0	77.5	76.2	35.0	36.1	45.0	35.6	46.3	90.8	63.2	92.6	67.0	40.9	92.3	38.1	52.5	50.1	41.8	57.2	60.3
CGNet [53]	95.5	78.7	88.1	40.0	43.0	54.1	59.8	63.9	89.6	67.6	92.9	74.9	54.9	90.2	44.1	59.5	25.2	47.3	60.2	64.8
ERFNet [11]	97.7	81.0	89.8	42.5	48.0	56.3	59.8	65.3	91.4	68.2	94.2	76.8	57.1	92.8	50.8	60.1	51.8	47.3	61.7	68.0
LEDNet [60]	98.1	79.5	91.6	47.7	49.9	62.8	61.3	72.8	92.6	61.2	**94.9**	76.2	53.7	90.9	64.4	64.0	52.7	44.4	71.6	70.6
FBSNet [15]	98.0	83.2	91.5	50.9	53.5	62.5	**67.6**	71.5	**92.7**	**70.5**	94.4	82.5	**63.8**	93.9	50.5	56.0	37.6	56.2	70.1	70.9
LARNet [67]	98.0	82.2	90.7	48.9	44.7	57.2	62.8	67.2	92.0	68.6	94.7	79.3	59.8	93.9	54.4	73.9	**61.3**	54.0	66.1	71.1
MSCFNet [14]	97.7	82.8	91.0	49.0	52.5	61.2	67.1	71.4	92.3	70.2	94.3	**82.7**	62.7	94.1	50.9	66.1	51.9	**57.6**	70.2	71.9
LETNet [39]	98.2	**83.6**	91.6	50.9	53.7	61.0	66.7	70.5	92.5	70.5	**94.9**	82.3	61.7	**94.4**	55.0	72.4	57.0	56.1	69.3	72.8
HAFormer (Ours)	**98.4**	82.6	**91.7**	**57.2**	**61.1**	**63.0**	62.2	**74.3**	91.8	61.7	93.8	79.3	56.4	93.7	**66.9**	**80.1**	65.3	56.7	**73.5**	**74.2**

ods is detailed in Table VIII. The experiments involve the spatial resolution of 512×1024 for evaluation, aligning with methods with the official code to ensure fairness. Table VIII demonstrates the impressive speed of HAFormer, achieving a frame rate of 105 fps when processing image streams of size 512×1024, positioning it as one of the fastest methods. While DABNet operates at 139 fps, HAFormer's competitive accuracy of 74.2% is significant for real-world applications like autonomous driving. Balancing speed (105 fps) and accuracy effectively, HAFormer emerges as a strong candidate for practical use.

V. Conclusions

In this study, we introduced HAFormer, a new lightweight semantic segmentation approach. We designed the Hierarchy-Aware Pixel-Excitation Module (HAPE) to extract enhanced hierarchical local features. Additionally, an Efficient Transformer module efficiently captures extensive global features with a limited computational load. Then, we incorporated a correlation-weighted Fusion (cwF) mechanism to combine highly correlated CNN and Transformer features for improved representational learning. Through extensive experiments on benchmark datasets, our approach has shown effectiveness and generalization, highlighting the capability of HAFormer to

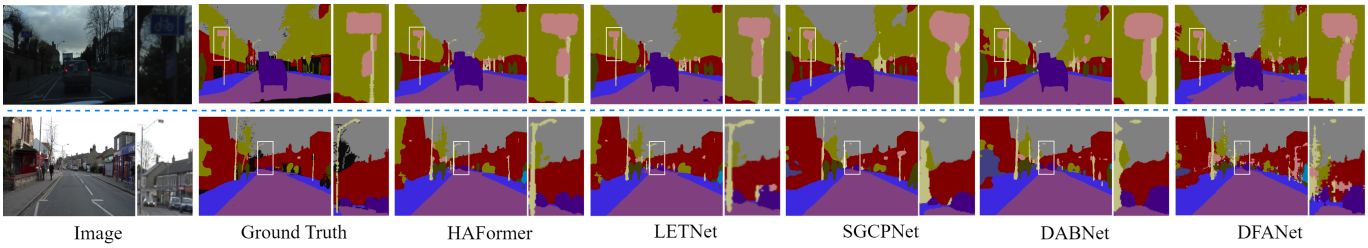

Fig. 10: Visual results obtained on the CamVid dataset. From left to right: original images, ground truths, predictions obtained with **HAFormer**, LETNet [39], SGCPNet [51], DABNet [52], DFANet [25]. Next to each prediction result is a partially enlarged detail map.

TABLE VII: Comparisons on the CamVid test dataset.

Methods	**Resolution**	**Backbone**	**Param. (M)** ↓	**GPU**	**Speed (FPS)** ↑	**mIoU (%)** ↑
ENet [10]	360 × 480	No	**0.36**	Titan X	98	51.3
SegNet [54]	360 × 480	VGG-16	29.50	Titan X	60	55.6
NDNet [58]	360 × 480	No	0.50	Titan X	78	57.2
DFANet [25]	720 × 960	Xception	7.80	Titan X	120	64.7
DABNet [52]	360 × 480	No	0.76	1080 Ti	136	66.4
ICNet [13]	720 × 960	PSPNet-50	7.80	Titan X	28	67.1
LARNet [67]	360 × 480	No	0.95	2080 Ti	204	67.1
EFRNet-16 [66]	720 × 960	EAA	1.44	Titan X	154	68.2
FBSNet [15]	360 × 480	No	0.62	2080 Ti	120	68.9
SGCPNet [51]	720 × 960	No	0.61	1080 Ti	**278**	69.0
MLFNet [61]	720 × 960	ResNet-34	13.03	Titan XP	57	69.0
MSCFNet [14]	360 × 480	No	1.15	Titan XP	110	69.3
AGLNet [68]	360 × 480	No	1.12	1080 Ti	90	69.4
LETNet [39]	360 × 480	No	0.95	3090	200	70.5
MGSeg [64]	736 × 736	ResNet-18	13.3	Titan XP	127	**72.7**
HAFormer (Ours)	360 × 480	No	0.60	2080 Ti	118	**71.1**
	720 × 960	No	0.60	2080 Ti	83	**71.9**

TABLE VIII: Comparisons of run-time and inference speed of the proposed HAFormer with other approaches.

Methods	Param. (M)↓	RTX 2080Ti 512 × 1024			
		FLOPs (G) ↓	ms ↓	fps ↑	mIoU (%) ↑
ENet [10]	**0.36**	3.8	13	112	58.3
NDNet [58]	0.50	**3.5**	8	119	61.1
DABNet [52]	0.76	10.5	**7**	**139**	69.3
FBSNet [15]	0.62	9.7	11	90	70.9
MSCFNet [14]	1.15	17.1	18	56	71.9
MLFNet [61]	13.03	15.5	14	72	72.2
LETNet [39]	0.95	13.6	11	91	72.8
SegTransConv [36]	7.00	10.2	17	58	73.0
HAFormer (ours)	0.60	11.1	10	105	**74.2**

achieve a balanced trade-off between segmentation accuracy and computational efficiency.

REFERENCES

[1] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 853–17 862.

[2] F. Yuan, Z. Zhang, and Z. Fang, "An effective CNN and transformer complementary network for medical image segmentation," *Pattern Recognition*, vol. 136, p. 109228, 2023.

[3] Y. Wang, H. Zhang, Y. Hu, X. Hu, L. Chen, and S. Hu, "Geometric boundary guided feature fusion and spatial-semantic context aggregation for semantic segmentation of remote sensing images," *IEEE Transactions on Image Processing*, vol. 32, pp. 6373–6385, 2023.

[4] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3146–3154.

[5] K. Li, Q. Geng, M. Wan, X. Cao, and Z. Zhou, "Context and spatial feature calibration for real-time semantic segmentation," *IEEE Transactions on Image Processing*, vol. 32, pp. 5465–5477, 2023.

[6] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3684–3692.

[7] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[8] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[9] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 593–602.

[10] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[11] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2017.

[12] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 552–568.

[13] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 405–

420.
[14] G. Gao, G. Xu, Y. Yu, J. Xie, J. Yang, and D. Yue, "MSCFNet: a lightweight network with multi-scale context fusion for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 489–25 499, 2022.
[15] G. Gao, G. Xu, J. Li, Y. Yu, H. Lu, and J. Yang, "FBSNet: A fast bilateral symmetrical network for real-time semantic segmentation," *IEEE Transactions on Multimedia*, vol. 25, pp. 3273–3283, 2023.
[16] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6881–6890.
[17] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.
[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
[19] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, "Unetr: Transformers for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 574–584.
[20] Y. Wu, K. Liao, J. Chen, J. Wang, D. Z. Chen, H. Gao, and J. Wu, "D-former: A u-shaped dilated transformer for 3d medical image segmentation," *Neural Computing and Applications*, vol. 35, no. 2, pp. 1931–1944, 2023.
[21] M. Heidari, A. Kazerouni, M. Soltany, R. Azad, E. K. Aghdam, J. Cohen-Adad, and D. Merhof, "Hiformer: Hierarchical multi-scale representations using transformers for medical image segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 6202–6212.
[22] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.
[23] Y. Lee, J. Kim, J. Willette, and S. J. Hwang, "MPViT: Multi-path vision transformer for dense prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7287–7296.
[24] Y. Yuan, R. Fu, L. Huang, W. Lin, C. Zhang, X. Chen, and J. Wang, "Hrformer: High-resolution vision transformer for dense predict," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7281–7293, 2021.
[25] H. Li, P. Xiong, H. Fan, and J. Sun, "Dfanet: Deep feature aggregation for real-time semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9522–9531.
[26] L. Li, T. Zhou, W. Wang, J. Li, and Y. Yang, "Deep hierarchical semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1246–1257.
[27] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "MViTv2: Improved multiscale vision transformers for classification and detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4804–4814.
[28] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
[29] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.
[30] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2881–2890.
[31] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "PVT v2: Improved baselines with pyramid vision transformer," *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, 2022.
[32] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.
[33] S. Jie and Z.-H. Deng, "FacT: Factor-tuning for lightweight adaptation on vision transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 1060–1068.
[34] W. Wang, W. Chen, Q. Qiu, L. Chen, B. Wu, B. Lin, X. He, and W. Liu, "CrossFormer++: A versatile vision transformer hinging on cross-scale attention," *arXiv preprint arXiv:2303.06908*, 2023.
[35] T. Yao, Y. Li, Y. Pan, Y. Wang, X.-P. Zhang, and T. Mei, "Dual vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10 870–10 882, 2023.
[36] J. Fan, B. Gao, Q. Ge, Y. Ran, J. Zhang, and H. Chu, "Segtransconv: Transformer and cnn hybrid method for real-time semantic segmentation of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 2, pp. 1586–1601, 2024.
[37] H. Wang, P. Cao, J. Wang, and O. R. Zaiane, "UCTransNet: rethinking the skip connections in u-net from a channel-wise perspective with transformer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 3, 2022, pp. 2441–2449.
[38] G. Gao, Z. Xu, J. Li, J. Yang, T. Zeng, and G.-J. Qi, "CTCNet: A CNN-transformer cooperation network for face image super-resolution," *IEEE Transactions on Image Processing*, vol. 32, pp. 1978–1991, 2023.
[39] G. Xu, J. Li, G. Gao, H. Lu, J. Yang, and D. Yue, "Lightweight real-time semantic segmentation network with efficient transformer and cnn," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 15 897–15 906, 2023.
[40] W. Zhang, Z. Huang, G. Luo, T. Chen, X. Wang, W. Liu, G. Yu, and C. Shen, "TopFormer: Token pyramid transformer for mobile semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 083–12 093.
[41] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 510–519.
[42] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 603–612.
[43] Y. Huang, W. Jia, X. He, L. Liu, Y. Li, and D. Tao, "Channelized axial attention for semantic segmentation," *arXiv preprint arXiv:2101.07434*, 2021.
[44] H. Yang and D. Yang, "CSwin-PNet: A CNN-Swin transformer combined pyramid network for breast lesion segmentation in ultrasound images," *Expert Systems with Applications*, vol. 213, p. 119024, 2023.
[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.
[46] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing convolutions to vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 22–31.
[47] Z. Lu, J. Li, H. Liu, C. Huang, L. Zhang, and T. Zeng, "Transformer for single image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 457–466.
[48] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.
[49] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
[50] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Proceedings of the European Conference on Computer Vision*. Springer, 2008, pp. 44–57.
[51] S. Hao, Y. Zhou, Y. Guo, R. Hong, J. Cheng, and M. Wang, "Real-time semantic segmentation via spatial-detail guided context propagation," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
[52] G. Li, I. Yun, J. Kim, and J. Kim, "DABNet: Depth-wise asymmetric bottleneck for real-time semantic segmentation," *arXiv preprint arXiv:1907.11357*, 2019.
[53] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, "CGNet: A light-weight context guided network for semantic segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 1169–1179, 2021.
[54] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.
[55] J. Gu, H. Kwon, D. Wang, W. Ye, M. Li, Y.-H. Chen, L. Lai, V. Chandra, and D. Z. Pan, "Multi-scale high-resolution vision transformer for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 094–12 103.
[56] H. Yan, C. Zhang, and M. Wu, "Lawin transformer: Improving semantic segmentation transformer with multi-scale representations via large window attention," *arXiv preprint arXiv:2201.01615*, 2022.
[57] Z. Lai, Y. Duan, J. Dai, Z. Li, Y. Fu, H. Li, Y. Qiao, and W. Wang, "Denoising diffusion semantic segmentation with mask prior modeling," *arXiv preprint arXiv:2306.01721*, 2023.
[58] Z. Yang, H. Yu, Q. Fu, W. Sun, W. Jia, M. Sun, and Z.-H. Mao, "NDNet: Narrow while deep network for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5508–5519, 2021.
[59] M. Liu and H. Yin, "Feature pyramid encoding network for real-time semantic segmentation," *arXiv preprint arXiv:1909.08599*, 2019.
[60] Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, and L. J. Latecki, "Lednet: A lightweight encoder-decoder network for real-time semantic segmentation," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1860–1864.
[61] J. Fan, F. Wang, H. Chu, X. Hu, Y. Cheng, and B. Gao, "Mlfnet: Multi-level fusion network for real-time semantic segmentation of autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 756–767, 2023.
[62] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *International Journal of Computer Vision*, vol. 129, pp. 3051–3068, 2021.
[63] Q. Lv, X. Sun, C. Chen, J. Dong, and H. Zhou, "Parallel complement network for real-time semantic segmentation of road scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4432–4444, 2022.
[64] J.-Y. He, S.-H. Liang, X. Wu, B. Zhao, and L. Zhang, "MGSeg: Multiple granularity-based real-time semantic segmentation network," *IEEE Transactions on Image Processing*, vol. 30, pp. 7200–7214, 2021.
[65] X. Liu, X. Shi, L. Chen, L. Qing, and C. Ren, "Efficient parallel multi-scale detail and semantic encoding network for lightweight semantic segmentation," in *Proceedings of the ACM International Conference on Multimedia*, 2023, pp. 2544–2552.
[66] X. Li, F. Yang, A. Luo, Z. Jiao, H. Cheng, and Z. Liu, "EFRNet: Efficient feature reconstructing network for real-time scene parsing," *IEEE Transactions on Multimedia*, vol. 24, pp. 2852–2865, 2022.
[67] X. Hu, S. Xu, and L. Jing, "Lightweight attention-guided redundancy-reuse network for real-time semantic segmentation," *IET Image Processing*, vol. 17, no. 9, pp. 2649–2658, 2023.
[68] Q. Zhou, Y. Wang, Y. Fan, X. Wu, S. Zhang, B. Kang, and L. J. Latecki, "AGLNet: Towards real-time semantic segmentation of self-driving images via attention-guided lightweight network," *Applied Soft Computing*, vol. 96, p. 106682, 2020.

Guoan Xu received the M.S. degree with the College of Automation and College of Artificial Intelligence, Nanjing University of Posts and Telecommunications. He is currently pursuing the Ph.D. degree with the Faculty of Engineering and Information Technology (FEIT), University of Technology Sydney (UTS). His main research interests include image segmentation, and multi-modality image processing.

Wenjing Jia received her Ph.D. degree in Computing Sciences from the University of Technology Sydney in 2007. She is currently an Associate Professor at the Faculty of Engineering and Information Technology (FEIT), University of Technology Sydney (UTS). Her research falls in the fields of image processing and analysis, computer vision, and pattern recognition.

Tao Wu received the BSc degree in Computer Science and Technology from Jilin University, Changchun, China, in 2020. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Nanjing University. His research interests include computer vision and deep learning.

Ligeng Chen received the BSc degree in Computer Science and Technology from Chongqing University, Chongqing, China, in 2017. He received his Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, Jiangsu, China, in 2023. He is currently a researcher and software engineer in Honor Device Co., Ltd. His research interests include data mining and binary code analysis.

Guangwei Gao (Senior Member, IEEE) received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology, Nanjing, in 2014. He is currently an Associate Professor in Nanjing University of Posts and Telecommunications. His research interests include pattern recognition and computer vision. He has published more than 70 scientific papers in IEEE TIP, IEEE TCSVT, IEEE TITS, IEEE TMM, PR, AAAI, IJCAI, etc. Personal website: *https://guangweigao.github.io*.