Boosting Semantic Segmentation from the Perspective of Explicit Class Embeddings

Yuhe Liu[1,2*] Chuanjian Liu[2] Kai Han[2†] Quan Tang[3] Zengchang Qin[1†]

[1]Beihang University [2]Huawei Noah's Ark Lab [3]South China University of Technology

{liuyuhe, zcqin}@buaa.edu.cn {liuchuanjian, kai.han}@huawei.com

csquantang@mail.scut.edu.cn

Abstract

Semantic segmentation is a computer vision task that associates a label with each pixel in an image. Modern approaches tend to introduce class embeddings into semantic segmentation for deeply utilizing category semantics, and regard supervised class masks as final predictions. In this paper, we explore the mechanism of class embeddings and have an insight that more explicit and meaningful class embeddings can be generated based on class masks purposely. Following this observation, we propose ECENet, a new segmentation paradigm, in which class embeddings are obtained and enhanced explicitly during interacting with multi-stage image features. Based on this, we revisit the traditional decoding process and explore inverted information flow between segmentation masks and class embeddings. Furthermore, to ensure the discriminability and informativity of features from backbone, we propose a Feature Reconstruction module, which combines intrinsic and diverse branches together to ensure the concurrence of diversity and redundancy in features. Experiments show that our ECENet outperforms its counterparts on the ADE20K dataset with much less computational cost and achieves new state-of-the-art results on PASCAL-Context dataset. The code will be released at https://gitee.com/mindspore/models and https://github.com/Carol-lyh/ECENet.

1. Introduction

Semantic segmentation is a fundamental task in computer vision, which aims to predict the corresponding classes for each pixel of the input image. Typically, pixels that share common semantic categories are aggregated together to form regions on each slice of predicted masks, which naturally presents the description of each category the model has learned.

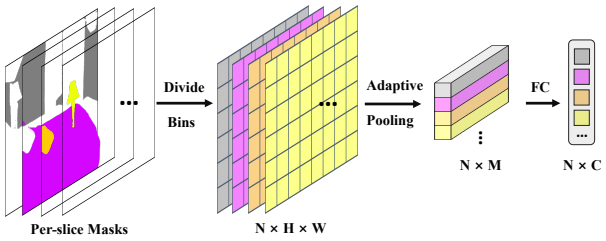

Figure 1. Explicit Class Extraction module. Each slice of predicted masks naturally presents what the model has learned on this category. By extracting class information into embeddings, we allow the information flow reversely and obtain the explicit class embeddings upon spatial prior knowledge on each region.

Traditional semantic segmentation methods are dominated by Fully Convolutional Networks (FCN) [33] based models. With stacked convolutional layers, the semantics in input images are gradually extracted. The 1×1 convolutional layer, which serves as semantic kernels, is usually applied to the representative feature maps in the end. Previous works [4, 5, 16, 30] focus on enlarging receptive filed [28, 4, 5], integrating attention modules [16, 51, 24] or fusing multi-stage features [34, 30, 27]. However, the CNN architectures are lack of long-range dependencies, which hinders the performances of FCNs.

Recently, transformer [43] using self-attention mechanism is introduced into the field of computer vision. Inspired by Vision Transformer (ViT) [13] and Segmenter [40], class tokens/embeddings have aroused the interest of many researchers [29, 10, 9, 56]. Generally, class embeddings are **randomly initialized** and passed into the decoder to interact with feature maps, then it would be used to get final segmentation masks [10, 9]. However, the class embeddings here are defined implicit and meaningless initially, which means that much spatial **prior knowledge** is ignored and lost.

*Work done during an internship at Huawei Noah's Ark Lab.
†Corresponding author.

Our key insight: class embeddings can be made **explicit** and **meaningful** by taking use of predicted masks. Intuitively, accurate regions on each slice of masks which the model has learned become the most natural description of each category. In fact, MaskFormer [10] first realizes the importance of per mask and replaces this per-pixel classification task with a set of binary masks prediction, each associated with a single category. Class embeddings which serve as object queries are passed into transformer decoder and then used to get the predicts. Given this promising attempt, a natural question emerges: *can we reverse this process? Or can predicted masks assist the generation of meaningful class embeddings conversely?*

To address this question, we consider a simple approach to utilize the predicted masks in turn and propose our ECENet, which is composed of Feature Reconstruction (FR), Explicit Class Extraction (ECE), Semantics Attention & Updater (SAU). More specifically, we design a FR module which is used to ensure the discriminative and informative capability of backbone features. Then Explicit Class Extraction (ECE) module consisting of spatial pooling and a single linear projection is used to extract class embeddings from masks. The class embeddings extracted are then passed into a transformer block that takes the early features as queries and class embeddings as keys and values. Through this, we encourage the interaction between early stage features and meaningful class semantics. Inspired by SegViT [53], we utilize the similarity maps as an extra masks. Then we employ gated mechanism [11] to further enhance previous class embeddings with the newly obtained semantics from the masks, dubbed Semantics Attention & Updater (SAU). Since the masks are a byproduct of the regular attention calculations, negligible computation is involved. We get the final segmentation masks by aggregating the enhanced outputs from multi-stage features and employ a convolutional layer onto it, assisted by class embeddings and intermediate masks.

Building upon this effective paradigm, the regions belonging to the same category tend to group together with the assistance of highly consistent semantics. Furthermore, the semantic gap between different layers is bridged. Experiments show that our ECENet achieves promising performance on common segmentation datasets and outperforms its counterparts with less computational cost.

We summarize our main contributions as follows:

- We reverse the general decoding process which departs from randomly initialized embeddings. Our class embeddings are explicit and consistently meaningful. It is the first attempt to uncover the correlations between segmentation masks and class embeddings and explore possible inverted information flow between both.

- We propose a new network, ECENet based on this insight, which is composed of Feature Reconstruction (FR), Explicit Class Extraction (ECE), Semantics Attention & Updater (SAU), which is demonstrated to be effective and efficient.

- We conduct extensive experiments on challenging benchmarks. Results show that our ECENet achieves competitive mIoU 55.2% on the ADE20K dataset with much less computational cost and parameters. We also demonstrate that our method yields **state-of-the-art** results on PASCAL-Context dataset (65.9% mIoU) and is compelling on Cityscapes dataset.

2. Related work

2.1. Semantic segmentation

Fully Convolutional Networks (FCN) [33] based models dominate the field of segmentation in early research. With stacked convolutional layers, the semantics in input images are gradually extracted. Meanwhile, the resolution of feature maps is reduced concerning computational cost and limited memory, which naturally forms hierarchical feature maps and broadens the receptive field. However, the inherent limited context information still hinders the performance of FCNs. To resolve this, many previous works focus on enlarging receptive filed [28, 4, 5] or integrating attention modules [16, 51, 24].

DeepLab [28] and DeepLabV2 [4] expand the receptive field by using dilated convolution. Despite this, an alternative approach is to integrate attention modules [16, 51, 24]. SENet [21] won the championship of the ImageNet 2017 image classification task by adding a channel attention mechanism to adjust the channel response adaptively. DANet [16] proposes the double attention network to simultaneously capture the global dependence in both spatial and channel dimension. [48] explores cross-image pixel contrast to focus on global context of the training data differently. Simultaneously, many approaches [24, 46, 15, 44] have been proposed to reduce the computational cost while retaining global attention.

2.2. Transformers for vision

Recently, transformer [43] architecture which can capture long-range dependencies has replaced CNNs as the new backbone to extract features. According to spatial size of the feature maps, transformer can be divided as plain [13] and hierarchical [32, 47, 50, 37] architectures. While the former remains the same resolution for all layers and the latter generally employs patch-merge methods between stages to get hierarchical resolutions.

Besides being used as a backbone, attention-based transformer structure is also designed as a decoder to extract high level semantic information. Segmenter [40], K-Net [56],

StructToken [29] and OneFormer [26] introduce learnable tokens or dynamic filters into Transformer decoder. Visual Parser [1] emphasizes the part-whole level attention and iteratively parses the two levels with the proposed encoder-decoder interaction. Further, MaskFormer [10] uncovers the importance of per mask and replaces the per-pixel classification task with a set of binary masks prediction, each associated with a single class. However, the information flow between class embeddings and predicted masks is one-way, from former to the latter. While much prior knowledge lying on accurate regions of each slice in masks is completely ignored and lost.

2.3. Multi-stage aggregation

Aggregating multi-stage features is an important method to improve recognition accuracy. Top-down feature fusion is used in [34, 20], which aims to optimize low-resolution features by using higher features stage by stage and eliminating the spatial information loss caused by down-sampling. DSSD [14], TDM [39] explored different approaches to improve feature aggregation, *e.g.* employing complex residual connections. FPN [30] proposes a feature pyramid architecture that combines multi-stage features via a top-down pathway and lateral connections. After that, PANet [31], NAS-FPN [17], BiFPN [41] and Recursive FPN [35] explore the geometric topology of the feature pyramid network to seek the optimum.

In contrast, we consider interacting and transferring semantic information between multi-stage features with the help of highly consistent class embeddings.

3. Methodology

In this section, we describe the proposed ECENet in detail. An overview of the model is shown in Fig. 3. The feature maps encoded from input images are briefly introduced in Section 3.1. To enhance the discriminability and informativity of features, we contribute a way to rebuild the features from backbone (Section 3.2). The proposed Explicit Class Extraction (ECE) and Semantics Attention & Updater (SAU), are described respectively in Section 3.3 and 3.4. And we conclude our whole model in Section 3.5.

3.1. Encoder

Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, it is transformed and reshaped into a sequence of tokens $\mathcal{F}_0 \in \mathbb{R}^{L\times C}$ where $L = HW/P^2$, P is the patch size and C is the number of channels. With positional information involved, the token sequence $\mathcal{F}_0$ is passed into 4 stages, each contains several transformer layers. Typically, there are patch merging modules between stages to gradually reduce the resolution of feature maps. After that, we obtain multi-level features at $\{1/4, 1/8, 1/16, 1/32\}$ of the original image resolution, which are defined as $[\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4]$. $\mathcal{F}_i \in \mathbb{R}^{L_i\times C_i}$ where

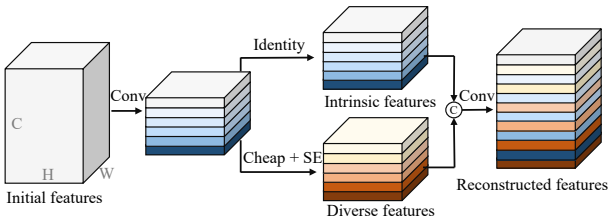

Figure 2. Feature Reconstruction module. It consists of two branches, named as intrinsic features and diverse features. By combining the two branches together, we ensure the **diversity** and **redundancy** simultaneously existing in features.

$L_i = H_i \times W_i$, represents the ith-stage feature whose scale is $1/2^{i+1}$ of the input image, *i.e.* $H_i = H/2^{i+1}$ and $W_i = W/2^{i+1}$, C_i is the embedding dimensions of stage i, $\forall i = 1, 2, 3, 4$.

It is worth noting that we introduce the typical hierarchical transformer backbone as our encoder here. However, our proposed method is also appropriate for plain ViT or the CNN backbone.

3.2. Feature reconstruction (FR)

The diversity and redundancy in feature maps is an important characteristic of successful networks [18], but has rarely been investigated, resulting in the unknown and hard-to-control feature representations. We point out that it is actually avoidable and can be improved.

To this end, we design a simple yet effective approach to purposefully control the diversity and redundancy of feature maps, dubbed Feature Reconstruction (FR). It can ensure better representation capabilities of features, thus assist in obtaining good-qualified semantic embeddings later. As observed in Fig. 2, we assume that the output feature map in each stage, $\mathcal{F}_i \in \mathbb{R}^{L_i\times C_i}$, which can be reshaped as $\mathcal{F}_i \in \mathbb{R}^{H_i\times W_i\times C_i}$, actually contains intrinsic features and we extract them using a 1x1 convolution followed by a norm layer.

$$\mathbf{Y}^{'} = \phi(\mathcal{F}_i) \in \mathbb{R}^{H_i\times W_i\times C_i//2} \tag{1}$$

To further ensure the diversity, we propose another diverse branch, which is realized by cheap linear operations on each intrinsic feature in $\mathbf{Y}^{'}$, following GhostNet [18]. However, we argue that this transform is uncontrollable, which indicates that the sensitivity to informative features is low. Thus we apply SE module [21] to re-calibrate channel-wise feature responses, ensuring that the useful ones are exploited more, as follows:

$$\mathbf{Y}^{''} = \mathrm{SE}(\mathrm{Cheap}(\mathbf{Y}^{'}) \in \mathbb{R}^{H_i\times W_i\times C_i//2} \tag{2}$$

Furthermore, the Softmax function is applied on each channel to exclusively concentrate on the most notable regions. Then all 'max' values on the feature maps are

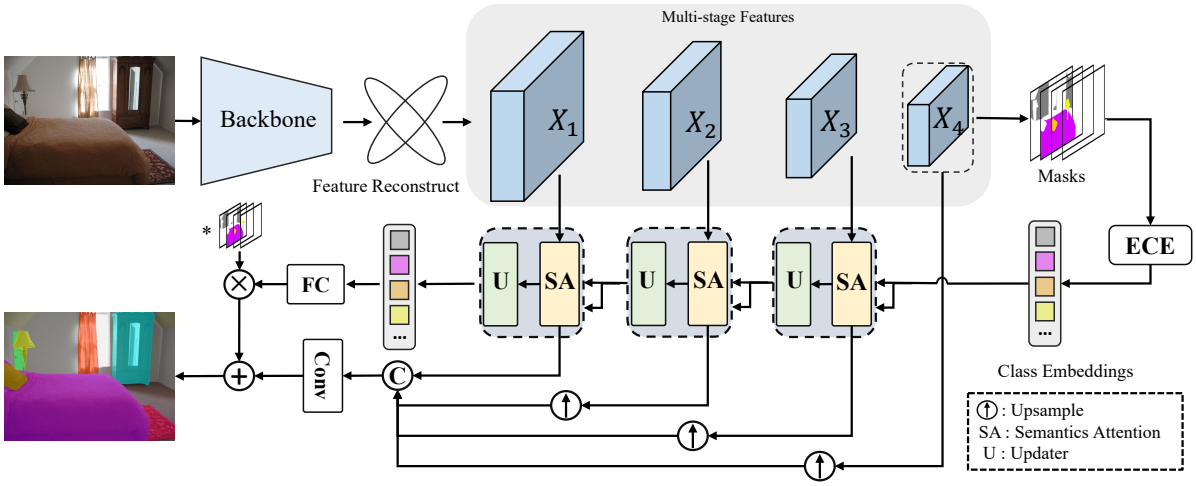

Figure 3. **The overall structure of ECENet.** It's composed of Feature Reconstruction (FR), Explicit Class Extraction (ECE), Semantics Attention & Updater (SAU). The FR module ensures the discriminative and informative capability of features from backbone. Then explicit class embeddings are generated from features of final stage by ECE module. After that, we carry out SAU module sequentially to make previous stage features interact with the class embeddings, thus higher-level semantics are transferred gradually. Masks emerge as a byproduct in the attention mechanism and are used to enhance our class embeddings. Finally, the enhanced multi-features are aggregated to get final predictions, assisted by class embeddings and summed masks. '*' means multiple masks.

summed which aims to ensure that each pixel is uniquely covered, especially for our pixel-level dense prediction tasks. Notably, we add minus in the summation results, namely diversity loss, $\mathcal{L}_{div}$. The formulation is:

$$L_{div}(\mathbf{Y}^{''}) = 1 - \frac{1}{\mathcal{C}} \sum_{k=1}^{HW} \max_{j=1,2,\cdots,\mathcal{C}} \frac{e^{\mathbf{Y}^{''}{}_{j,k}}}{\sum_{k'=1}^{WH} e^{\mathbf{Y}^{''}{}_{j,k'}}} \tag{3}$$

where $\mathcal{C}$ equals to half of the channels. Here we emphasis that in dense prediction tasks, each pixel should be noticed, thus making each slice of feature maps unique is reasonable. This process requires almost no extra computation. And it can improve the performance by a large margin, which is further shown in ablation experiments.

Finally, the intrinsic branch $\mathbf{Y}^{'}$ and diverse branch $\mathbf{Y}^{''}$ are concated together and projected to the initial shape. By this way, we ensure the diversity and redundancy simultaneously existing in each stage's representative feature maps.

3.3. Explicit class extraction (ECE)

Intuitively, accurate regions on each slice of predicted masks become the most natural description of each category. With this insight, we seek to get explicitly defined class embeddings by taking use of predicted masks.

Specifically, after feature reconstruction, we unify their channels by a 1x1 convolution and define the unified features as $[\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4]$, where $\mathcal{X}_i \in \mathbb{R}^{C \times H_i \times W_i}$, $\forall i = 1, 2, 3, 4$, each holds {1/4, 1/8, 1/16, 1/32} of the original image resolution. C is the unified dimension, usually set to 256. Then, linear transformations are applied to the feature map $\mathcal{X}_4$ from last stage, as presented by Eq. (4).

$$Mask(\mathcal{X}_4) = \phi_2(\phi_1(\mathcal{X}_4)) \tag{4}$$

where ϕ_1 and ϕ_2 are linear transformations implemented by 1×1 convolutional layers without activation. $Mask(\mathcal{X}_4) \in \mathbb{R}^{N \times H_4 \times W_4}$ becomes the intermediate masks with N equals to the number of classes.

Next, the Explicit Class Extraction module is employed, as illustrate in Fig. 1. Specifically, each slice of the masks is divided into spatial bins. Then we conduct parameter-free pooling in each spatial bin to harvest sub-region representations.

But this operation is non-trivial. We aim to cover class information on different sized areas. Inspired by [19], we use spatial pyramid pooling to maintain both local and global information. Intuitively, the biggest divided number of a side ought to depend on datasets. Thus the maximum pooling size is set to be proportional to $\sqrt{N}$, *i.e.* $\alpha\sqrt{N}$, where N is the total number of classes. Ratio α is adjusted to different datasets. Then a single linear projection just follows to convert the channels to unified dimension C and obtain the explicit class embeddings $\mathcal{G} \in \mathbb{R}^{N \times C}$. The whole process can be summarized in Eq. (5).

$$\mathcal{G} = \psi(\text{Pooling}(Mask(\mathcal{X}_4))) \tag{5}$$

where ψ is a simple linear transformation. Through ECE module, we bridge the gap between segmentation masks and class embeddings, allow the information flow reversely and obtain the explicit class embeddings instead of random initialized blindly, which completes the last piece of the puzzle in recent segmentation tasks.

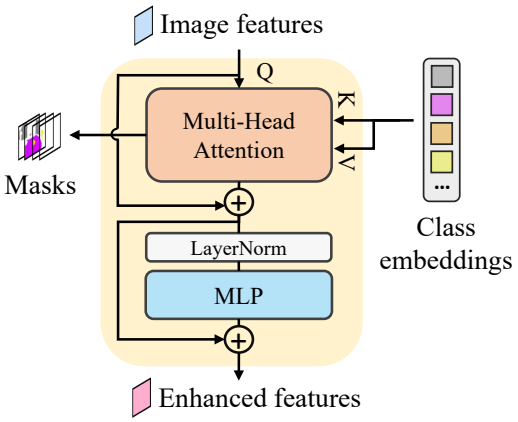

Figure 4. Semantics Attention module (SA). Image features are strengthened during interacting with explicit class embeddings from higher stages. Note that the LayerNorm applied on image and class embeddings is omited for simplicity.

3.4. Semantics attention & updater (SAU)

This part integrates a Semantics attention module used for interacting between image features and explicit class embeddings, and an update head which refreshes the class representations using the newly obtained ones.

Semantics Attention. As illustrated in Fig. 4, the Semantics Attention module consists of a multi-head attention (MSA) block and a MLP block with residual connections added after every block. Specifically, it takes earlier features $\mathcal{X}_{i-1}$ as queries and class embeddings $\mathcal{G} \in \mathbb{R}^{N\times C}$ as keys and values. Firstly, linear transformations are applied to form query (Q), key (K) and value (V), as presented by Eq. (6).

$$\begin{aligned} Q &= \phi_q(\mathcal{X}_{i-1}) \in \mathbb{R}^{L_{i-1}\times C}, \\ K &= \phi_k(\mathcal{G}) \in \mathbb{R}^{N\times C}, \\ V &= \phi_v(\mathcal{G}) \in \mathbb{R}^{N\times C}. \end{aligned} \tag{6}$$

Following the scaled dot-product attention mechanism, Q and K are interacted to measure Class-Feature similarity. The similarity map and attention map are calculated as following:

$$\begin{aligned} S(Q,K) &= \frac{QK^T}{\sqrt{d_k}}, \\ \mathrm{MSA}(\mathcal{G},\mathcal{X}_{i-1}) &= \mathrm{Softmax}(S(Q,K))V. \end{aligned} \tag{7}$$

where $\sqrt{d_k}$ serves as a scaling factor while d_k equals to the dimension of keys, $S(Q,K) \in \mathbb{R}^{L_{i-1}\times N}$, and $\mathrm{MSA}(\mathcal{G}, \mathcal{X}_{i-1}) \in \mathbb{R}^{L_{i-1}\times C}$. With the guidance of clearly defined class embeddings, the regions belonging to the same category tend to group together and return strengthened representative features.

Then a MLP block of three layers is applied. Note that, we follow [50] to employ a 3×3 depth-wise convolution which considers the effects of zero padding to leak position information. LayerNorm (LN) is applied before every block.

Class Updater. The clearly defined class embeddings initially emerge from features of final stage. However, in segmentation tasks, models need to recognize regions that vary in scale, which call for multi-scale spatial information enhancement. Same goes for our explicit class embeddings. Fortunately, the accurate masks appear in an inner way, as a byproduct where Q and K interact to measure Class-Feature similarity. Inspired by [53], we reshape and utilize this similarity map as new generated masks, where $S(Q,K) \in \mathbb{R}^{L_{i-1}\times N}$, to $Mask(\mathcal{X}_{i-1}, \mathcal{G}) \in \mathbb{R}^{N\times H_{i-1}\times W_{i-1}}$. After that, we employ Explicit-Class-Extraction (ECE) module to get newly obtained class embeddings $\hat{\mathcal{G}}$:

$$\hat{\mathcal{G}} = \mathrm{ECE}(Mask(\mathcal{X}_{i-1}, \mathcal{G})) \tag{8}$$

Then gated mechanism is applied to further refresh previous class embeddings. Specifically, we first fuse the new class representations $\hat{\mathcal{G}}$ with previous ones as

$$F_u = \phi_3(\hat{\mathcal{G}}) \odot \phi_4(\mathcal{G}) \tag{9}$$

where $F_u \in \mathbb{R}^{N\times C}$, ϕ_3 and ϕ_4 are linear transformations. Then an updating gate, U is learned and multiplied with the projected $\hat{\mathcal{G}}$. Finally, we add this with the previous $\mathcal{G}$ to get the updated class embeddings, as follows:

$$\begin{aligned} U &= \sigma(\psi_1(F_u)) \\ \mathcal{G} &= U \odot \psi_2(\hat{\mathcal{G}}) + \mathcal{G} \end{aligned} \tag{10}$$

where σ is the Sigmoid function and ψ_1, ψ_2 are fully connected (FC) layers followed by LayerNorm (LN), as in [56]. This Semantics Attention & Updater module repeats progressively on the remaining multi-stage features, resulting highly informative class embeddings. Furthermore, the representative image features are enhanced during Class-Feature interaction, eliminating semantic gap between different layers.

3.5. The ECENet structure

As illustrated in Fig. 3, we get reconstructed features $[\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3, \mathbf{Y}_4]$ from FR module using multi-stage features and unify their channels to get $[\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4]$. Then explicit class embeddings is extracted from last feature $\mathcal{X}_4$. Previous stage features interact with the class embeddings and higher-level semantics are transferred gradually. Masks emerge as a byproduct in the attention mechanism and are used to enhance our class embeddings.

Finally, the enhanced multi-stage features are upsampled using ghost feature [18] and pixel shuffle [38]. Then all features are concated together and pass through a

1x1 convolution to get the final prediction. As in DETR [3] and SegViT [53], we apply a linear transformation followed by a Softmax activation to the final class embeddings to get class probability predictions. Then it multiplies with summed masks, and added on to enhance the final prediction. Formally, the loss function can be formulated as:

$$\mathcal{L}_{overall} = \mathcal{L}_{cls} + \mathcal{L}_{mask} + \lambda_{div}\mathcal{L}_{div} \tag{11}$$

where

$$\mathcal{L}_{mask} = \lambda_{focal}\mathcal{L}_{focal} + \lambda_{dice}\mathcal{L}_{dice} \tag{12}$$

The classification loss ($\mathcal{L}_{cls}$) is implemented by Cross-Entropy loss and the masks are summed together, supervised by the mask loss ($\mathcal{L}_{mask}$) which is a linear combination of a focal loss and a dice loss multiplied by hyper-parameters λ_{focal} and λ_{dice} respectively as in DETR [3]. The $\mathcal{L}_{div}$ is applied on each stage's 'diverse' branch features. The experiments in next section show that this design is efficient and works well.

Different from Existing Methods. Though this mask-to-class transform is the converse of traditional prediction process, there are still some works that vaguely realize the importance of this. In CNN networks, ACFNet [54] tried to perform matrix multiplication between feature maps $F \in \mathbb{R}^{C\times H\times W}$ and coarse mask $P \in \mathbb{R}^{N\times H\times W}$. However, it faces challenges if only a pure mask is given. Instead, we emphasize the predicted mask itself and reveal the underlying mechanism. The accurate regions on predicted masks present each category independently, without the assistance of feature maps F. Actually, if extra feature maps are brought in to get class embeddings, it actually confuses the purely meaningful region representations in masks.

Further, our method not only improves the efficiency, but also has better interpretability by incorporating class information into embeddings, as shown later in Section 4.4. Basically, both our method and Mask2Former [9] recognize and utilize the value in masks which localize regions, but with different methods. Mask2Former [9] uses 100+ queries and masked-attention that aims to focus on specific regions. While we have new insights on masks and transfer these into Explicit Class Embeddings, which is more explainable. And this idea is well suited to other frameworks.

Besides, we want to emphasize that though good performance is achieved in some works, *e.g.* K-Net [56] which introduces dynamic class kernels, however, the boosting in performance actually is lying in the usage of supervised masks to polish the learned kernels gradually.

4. Experiments

4.1. Datasets

ADE20K is a scene parsing dataset covering 150 fine-grained semantic concepts, consisting of $20,210$ images as the training set and $2,000$ images as the validation set.

PASCAL-Context contains $4,996$ and $5,104$ images for training and validation respectively. Following previous works, we evaluate on the most frequent 59 classes.

Cityscapes is a driving-scene dataset densely annotates 19 object categories in images. It contains $5,000$ finely annotated images, split into $2,975/500/1,525$ for training, validation and testing respectively.

4.2. Implementation details

Training settings We mainly use the Swin Transformer as the backbone. Specifically, we provide results primarily on its 'Large' variation and use its 'Base' variation for most ablation studies. Our experiments are based on MMSegmentation [12] and follow the commonly used training settings. During training, we use AdamW as the optimizer with a total iteration of $160k$, $80k$ and $160k$ for ADE20K, PASCAL-Context and Cityscapes respectively. The batch size is set to 16 except that we use batch size 8 for Cityscapes. Ratio α is empirically set to 1, $\sqrt{2}$, 3 for ADE20K, PASCAL-Context and Cityscapes respectively. We employ data augmentation sequentially via random resize with the ration between 0.5 and 2.0, random horizontal flipping, and random cropping (640×640 for ADE20K, 480×480 for PASCAL-Context and 1024×1024 for Cityscapes).

Evaluation metric. We use the mean intersection over union (mIoU) to evaluate the segmentation performance. All reported mIoU scores are in a percentage format. 'ss' means single-scale testing and 'ms' means test time augmentation with multi-scaled $(0.5, 0.75, 1.0, 1.25, 1.5, 1.75)$ inputs. All reported number of parameters (Params) and computational costs in GFLOPs are measured using the fvcore [1] library.

4.3. Comparison with State-of-the-Arts

Results on ADE20K. Table 1 reports comparison with the state-of-the-art methods on ADE20K validation set. Our ECENet outperforms other counterparts by a large margin on diverse backbones with relatively small amount of parameters. With the SwinT-Large backbone, our ECENet achieves 55.2% mIoU using only 208.3M paramenters and 425.7 GFLOPs. It is 1.0% mIoU better than the recent SenFormer [2] using the same backbone. Noting that, the parameters and computational cost of our method is much less than others. Our method based on SwinT-Base backbone achieves 53.4% mIoU under single-scale inference, which is similar to SenFormer [2] (53.1% mIoU) based on SwinT-Large and UPerNet + ViT-Adapter [7] (53.4% mIoU) based on ViT-Large, but with much less parameters (96.7M vs. 364M) and computational cost (243.3 GFLOPs vs. 546 GFLOPs).

[1] https://github.com/facebookresearch/fvcore

Method	Backbone	Crop Size	GFLOPs	#param.	mIoU (ss)	mIoU(ms)
Zhou et al. [59]	ResNet-101	512 × 512	-	68.5M	41.1	-
PSPNet [57]	ResNet-101	512 × 512	257	65.7M	44.4	45.4
UPerNet [49]	ResNet-101	512 × 512	258	85.5M	43.8	44.9
ECENet (Ours)	ResNet-101	512 × 512	293.1	60.2M	**45.3**	**46.8**
DPT [36]	ViT-Base	512 × 512	219.8	-	47.2	47.9
StructToken-SSE [29]	ViT-Base	512 × 512	>150	142M	50.9	51.8
UPerNet + SwinT [32]	SwinT-Base†	640 × 640	471	121.4M	-	51.6
ECENet (Ours)	SwinT-Base†	640 × 640	243.3	96.7M	**53.4**	**54.2**
SETR-PUP [58]	ViT-Large	640 × 640	711	308M	48.2	50.0
Segmenter [40]	ViT-Large	640 × 640	672	333M	51.7	53.6
StructToken-CSE [29]	ViT-Large†	640 × 640	>700	350M	52.8	54.2
UPerNet + ViT-Adapter [7]	ViT-Large†	640 × 640	-	364M	53.4	54.4
UPerNet + SwinT [32]	SwinT-Large†	640 × 640	647	234M	-	53.5
UperNet + KNet [56]	SwinT-Large†	640 × 640	659	245M	-	54.3
SenFormer [2]	SwinT-Large†	640 × 640	546	233M	53.1	54.2
ECENet (Ours)	SwinT-Large†	640 × 640	425.7	208.3M	**54.1**	**55.2**

Table 1. Experiment results on the ADE20K `val.` split. '†' means the model's weight are pretrained on ImageNet-22K. The GFLOPs is measured at single-scale inference with the given crop size. Note that we also use dilated ResNet-101 as backbone.

Method	Backbone	GFLOPs	mIoU(ms)
PSPNet [57]	ResNet-101	157	47.8
EncNet [55]	ResNet-101	192.1	52.6
HRNetv2 [45]	HRNetv2-W48	82.7	54.0
NRD [52]	ResNet-101	42.9	54.1
CAA [23]	ResNet-101	-	55.0
SegViT (Shrunk) [53]	ViT-Large	186.9	63.7
SegViT [53]	ViT-Large	321.6	65.3
UPerNet + CAR [22]	SwinT-Large†	-	59.0
SenFormer [2]	SwinT-Large†	>300	64.5
ECENet (Ours)	SwinT-Large†	241.9	**65.9**

Table 2. Experiment results on PASCAL-Context validation set with multi-scale inference. '†' means the model's weight are pretrained on ImageNet-22K. The GFLOPs is measured at single-scale inference with a crop size of 480 × 480.

Method	Backbone	mIoU(ms)
PSPNet [57]	ResNet-101	78.5
DeepLabv3+ [6]	Xception-71	79.6
CCNet [24]	ResNet-101	81.3
CANet [42]	ResNet-101	81.9
AlignSeg [25]	ResNet-101	82.4
P-DeepLab [8]	Xception-71	81.5
SegFormer [50]	MiT-B5	84.0
SETR-PUP [58]	ViT-Large	82.2
Segmenter [40]	ViT-Large	81.3
StructToken-PWE [29]	ViT-Large	82.1
Mask2Former [9]	SwinT-Large†	84.3
ECENet (Ours)	SwinT-Large†	**84.5**

Table 3. Experiment results on Cityscapes validation set with multi-scale inference. '†' means the model's weight are pretrained on ImageNet-22K.

Results on PASCAL-Context. Table 2 shows the result on PASCAL-Context dataset. We follow SenFormer [2] to evaluate our method and report the results under 59 classes. Our ECENet achieves mIoU 65.9%, which outperforms the recent SenFormer [2] with the same SwinT-Large backbone by 1.4% mIoU, and achieves new state-of-the-art performance. Compared with SegViT [53], our ECENet outperforms it by 0.6% mIoU with much less computational cost (reduced 25% GFLOPs).

Results on Cityscapes. Table 3 shows the result on Cityscapes validation set. Our method reaches mIoU 84.5%, surpassing Mask2Former [9] by 0.2% mIoU, which is very competitive with the previous works.

4.4. Ablation study

In this section, we conduct experiments on ADE20K dataset with SwinT-Base backbone to show the effectiveness of our proposed ECENet method.

Ablation of the components in ECENet Table 4 shows the effect of different components in ECENet. Dynamic tokens means that the class embeddings are updated by masks which is a byproduct of attention. We can see that Feature Reconstruction is capable of providing 0.6% mIoU of performance promotion. It's more beneficial to make use of the diversity loss, $\mathcal{L}_{div}$ to make sure the diversity and redundancy in feature maps before interaction.

Ablation of loss coefficient Table 5 shows the ablation of loss coefficient λ_{div}. The adopted choice of loss coeffi-

Figure 5. Competitive segmentation results on the ADE20K, PASCAL-Context and Cityscapes validation set.

Dynamic tokens	FR(wo $\mathcal{L}_{div}$)	$\mathcal{L}_{div}$	mIoU (ss)	GFLOPs
✓			52.5	240.7
✓	✓		53.1	243.3
✓	✓	✓	53.4	243.3

Table 4. Ablation results on the components in ECENet. The experiment is carried on ADE20K dataset with SwinT-Base backbone. The crop size is 640×640. 'FR': Feature Reconstruction.

λ_{div}	mIoU (ss)
0.1	53.25
0.2	**53.40**
0.5	52.65

Table 5. Ablation results on loss coefficient λ_{div}. The experiments use the SwinT-Base backbone and are carried out on ADE20K dataset.

Updater	mIoU (ss)
plus	52.7
gated	**53.4**

Table 6. Ablation results on the updating method of the Updater on ADE20K dataset using SwinT-Base as backbone.

cient λ_{div} is purely result-driven. We compare the model performance under 3 levels of loss coefficient, 0.1, 0.2, 0.5. Experiments show that our ECENet achieves the optimal 53.4% mIoU when $\lambda_{div} = 0.2$. We thus use this setting for all experiments.

Ablation of the updating method. To demonstrate the effectiveness of our class updater, we conduct comparison between the naive plus and gated operation, observed in Table 6 . Experiments show that our gated operation surpasses the naive plus by 0.7% mIoU.

Applying to other methods. Since we get the insight that supervised masks could serve as the best tool to get class embeddings, we wish to study the effect of our explicit class embeddings obtained from masks when cooperating with randomly-initialized ones. Thus we conduct ablation studies on the recent SegViT [53]. As shown in Table 7, the application of our gated-updating on the original randomly initialized class embeddings could also provide a performance boost by a large margin of 0.6% mIoU, which shows our potential on different structures and methods. Besides, the gain on the computational cost is negligible.

Method	mIoU (ss)	GFLOPs
original	51.3	120.9
gated-updating	**51.9**	121.2

Table 7. Verification of our gated-updating strategy using SegViT on ADE20K. The backbone is ViT-Base. The GFLOPs are measured at single-scale inference with a crop size of 512×512.

4.5. Visualization

The visualization results of our ECENet with SwinT-Large as the backbone are shown in Fig. 5. Notably, our method can produce satisfactory segmentation results in various indoor and outdoor scenes on ADE20K, PASCAL-Context and Cityscapes validation set, especially at the edges of objects, *e.g.* trees, leaves and chairs. This demonstrates that our model can achieve extraordinary promotion in exploiting implications of class embeddings. There is great potential in exploring the flow of information from masks to class embeddings and we can gain a deeper understanding of the category semantics learned by the model.

5. Conclusion

This work utilizes explicit class embeddings to boost the performance of semantic segmentation. In contrast to the existing randomly initialized class embeddings which are content-ignored and implicit, we generate them with the coarsely predicted segmentation masks, which makes them consistent and meaningful initially. Furthermore, with the guidance of clearly defined class embeddings, the regions in feature maps which belong to the same category tend to group together and return strengthened representative features. The explicit class embeddings also alleviate the semantic gap between different layers. Besides, we also propose a novel Feature Reconstruction module that ensure the discriminability and informativity of features from backbone. Our method achieves the new state-of-the-art performance with less computational cost on PASCAL-Context. Notably, we believe this can spark more interest in revealing the true meanings behind the category semantics.

Acknowledgment This research was supported by Huawei Noah's Ark Lab. We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research. In particular, we would also like to thank Yifan Liu, Ning Ding and Bowen Zhang for the help and suggestions at the early stage.

References

[1] Song Bai, Philip Torr, et al. Visual parser: Representing part-whole hierarchies with transformers. *arXiv preprint arXiv:2107.05790*, 2021. 3

[2] Walid Bousselham, Guillaume Thibault, Lucas Pagano, Archana Machireddy, Joe Gray, Young Hwan Chang, and Xubo Song. Efficient self-ensemble framework for semantic segmentation. *arXiv preprint arXiv:2111.13280*, 2021. 6, 7

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 6

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1, 2

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint*, 2017. 1, 2

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 7

[7] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 6, 7

[8] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12485, 2020. 7

[9] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 1, 6, 7

[10] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 1, 2, 3

[11] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2

[12] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 6

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021. 1, 2

[14] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 3

[15] Jun Fu, Jing Liu, Jie Jiang, Yong Li, Yongjun Bao, and Hanqing Lu. Scene segmentation with dual relation-aware attention network. *IEEE Transactions on Neural Networks and Learning Systems*, 32(6):2547–2560, 2020. 2

[16] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual Attention Network for scene segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019. 1, 2

[17] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7036–7045, 2019. 3

[18] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1580–1589, 2020. 3, 5

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 4

[20] Sina Honari, Jason Yosinski, Pascal Vincent, and Christopher Pal. Recombinator networks: Learning coarse-to-fine feature aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5743–5752, 2016. 3

[21] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2, 3

[22] Ye Huang, Di Kang, Liang Chen, Xuefei Zhe, Wenjing Jia, Xiangjian He, and Linchao Bao. Car: Class-aware regularizations for semantic segmentation. *arXiv preprint arXiv:2203.07160*, 2022. 7

[23] Ye Huang, Di Kang, Wenjing Jia, Liu Liu, and Xiangjian He. Channelized axial attention–considering channel relation within spatial attention for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1016–1025, 2022. 7

[24] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 603–612, 2019. 1, 2, 7

[25] Zilong Huang, Yunchao Wei, Xinggang Wang, Wenyu Liu, Thomas S Huang, and Humphrey Shi. Alignseg: Feature-aligned segmentation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):550–557, 2022. 7

[26] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2989–2998, 2023. 3

[27] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019. 1

[28] Chen Liang-Chieh, George Papandreou, Iasonas Kokkinos, kevin murphy, and Alan Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015. 1, 2

[29] Fangjian Lin, Zhanhao Liang, Junjun He, Miao Zheng, Shengwei Tian, and Kai Chen. Structtoken: Rethinking semantic segmentation with structural prior. *arXiv preprint arXiv:2203.12612*, 2022. 1, 3, 7

[30] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 3

[31] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 3

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 7

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1, 2

[34] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 75–91. Springer, 2016. 1, 3

[35] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10213–10224, 2021. 3

[36] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 7

[37] Sucheng Ren, Daquan Zhou, Shengfeng He, Jiashi Feng, and Xinchao Wang. Shunted self-attention via multi-scale token aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10853–10862, 2022. 2

[38] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 5

[39] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*, 2016. 3

[40] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021. 1, 2, 7

[41] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 3

[42] Quan Tang, Fagui Liu, Tong Zhang, Jun Jiang, Yu Zhang, Boyuan Zhu, and Xuhao Tang. Compensating for local ambiguity with encoder-decoder in urban scene segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 2022. 7

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 2

[44] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, pages 108–126. Springer, 2020. 2

[45] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 7

[46] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020. 2

[47] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 2

[48] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7303–7313, 2021. 2

[49] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 7

[50] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 2, 5, 7

[51] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 1, 2

[52] Bowen Zhang, Zhi Tian, Chunhua Shen, et al. Dynamic neural representational decoders for high-resolution semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17388–17399, 2021. 7

[53] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, et al. Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems*, 35:4971–4982, 2022. 2, 5, 6, 7, 8

[54] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6798–6807, 2019. 6

[55] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018. 7

[56] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *Advances in Neural Information Processing Systems*, 34:10326–10338, 2021. 1, 2, 5, 6, 7

[57] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 7

[58] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS 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*, pages 6881–6890, 2021. 7

[59] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2582–2593, 2022. 7