

PP-MobileSeg: Explore the Fast and Accurate Semantic Segmentation Model on Mobile Devices

Shiyu Tang* Ting Sun* Juncai Peng* Guowei Chen Yuying Hao
 Manhui Lin Zhihong Xiao Jiangbin You Yi Liu
 Baidu Inc.

{tangshiyu, suntling13, pengjuncai}@baidu.com

Abstract

The success of transformers in computer vision has led to several attempts to adapt them for mobile devices, but their performance remains unsatisfactory in some real-world applications. To address this issue, we propose PP-MobileSeg, a semantic segmentation model that achieves state-of-the-art performance on mobile devices. PP-MobileSeg comprises three novel parts: the StrideFormer backbone, the Aggregated Attention Module (AAM), and the Valid Interpolate Module (VIM). The four-stage StrideFormer backbone is built with MV3 blocks and strided SEA attention, and it is able to extract rich semantic and detailed features with minimal parameter overhead. The AAM first filters the detailed features through semantic feature ensemble voting and then combines them with semantic features to enhance the semantic information. Furthermore, we proposed VIM to upsample the downsampled feature to the resolution of the input image. It significantly reduces model latency by only interpolating classes present in the final prediction, which is the most significant contributor to overall model latency. Extensive experiments show that PP-MobileSeg achieves a superior tradeoff between accuracy, model size, and latency compared to other methods. On the ADE20K dataset, PP-MobileSeg achieves 1.57% higher accuracy in mIoU than SeaFormer-Base with 32.9% fewer parameters and 42.3% faster acceleration on Qualcomm Snapdragon 855. Source codes are available at <https://github.com/PaddlePaddle/PaddleSeg/tree/release/2.8>.

1. Introduction

Semantic segmentation is a computationally expensive task compared to other computer vision tasks like image classification [18] or object detection [37], as it involves

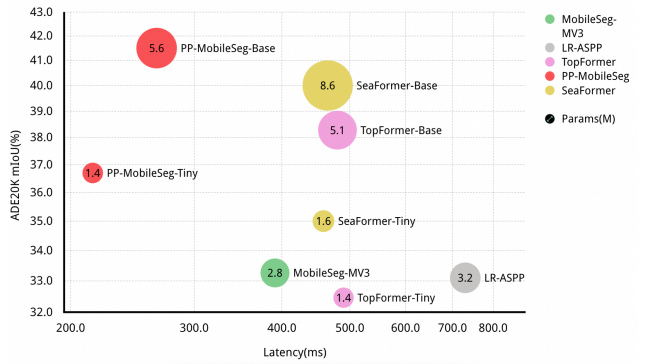


Figure 1. We present the accuracy-latency-params analysis of our proposed PP-MobileSeg model on the ADE20K validation set. The trade-off analysis is represented as a bubble plot, where the x-axis denotes the latency and the y-axis denotes the mIoU. Models with the same color are from the same model series. Our model achieves a better accuracy-latency-params trade-off. Note that the latency is tested with the final ArgMax operator using PaddleLite on Qualcomm Snapdragon 855 CPU with a single thread and 512x512 as input shape.

predicting the class of every pixel. While there have been significant advancements in semantic segmentation on GPU devices, few studies have addressed the challenges of mobile semantic segmentation [12, 27, 33]. This lack of research impedes the practical application of semantic segmentation to mobile applications.

Recently, the surge of vision transformers (ViTs) [10] proved the promising performance of transformer-based neural networks on semantic segmentation [5, 24, 31, 35]. Various works have proposed transformer-CNN hybrid architectures for lightweight neural network design, such as MobileViT [21], MobileFormer [4], and EdgeNext [20]. This hybrid architecture combines global and local information in neural networks at the lowest possible cost. However, the computation complexity of Multi Head Self Attention (MHSA) makes these networks hard to be deployed

*These authors contributed equally to this work

on mobile devices. Even though several efforts have been made to decrease the time complexity, including shifted window attention [16], efficient attention [23], external attention [29], axial attention [28], SEA attention [27] and etc. But many of these techniques require complex index operations that ARM CPUs cannot support [27]. Besides latency and accuracy, memory storage is also a crucial element for mobile applications, since memory storage is limited on mobile devices. Therefore the fundamental question arises: *can we design a hybrid network for mobile devices with a superior trade-off between parameter, latency, and accuracy?*

In this work, we address the above question by exploring the mobile segmentation architecture under the constraints of model size and speed for a performance leap forward. Under extensive search, we manage to propose three novel designed modules: the four-stage backbone StrideFormer, the feature fusion block AAM, and the upsample module VIM as shown in Fig. 2. By combining these modules, we propose a family of SOTA mobile semantic segmentation networks called PP-MobileSeg, which is well-suited for mobile devices with great parameters, latency, and accuracy balance. Our improved network design allows PP-MobileSeg-Base to improve the inference speed by 40% and 34.9% less model size than SeaFormer while maintaining a competitive 1.37 higher mIoU (Tab. 1). Compared with MobileSeg-MV3, PP-MobileSeg-Tiny achieves 3.13 higher mIoU while being 45% faster and 49.5% smaller (Tab. 1). We also evaluate the performance of PP-MobileSeg on the Cityscapes dataset [6] (Tab. 2), which shows its superiority in model performance on high-resolution inputs. Although PP-MobileSeg-Base has slightly longer latency, it maintains model size superiority while being 1.96 higher in mIoU than SeaFormer [27] on the cityscapes dataset [6].

In summary, our contributions are as follows:

- We introduce the StrideFormer, a four-stage backbone with MobileNetV3 blocks that efficiently extracts features of different receptive fields while minimizing parameter overhead. We also apply strided SEA attention [13, 27] to the output of the last two stages to improve global feature representation under computation constraints.
- We propose the Aggregate Attention Module (AAM), which fuses features from the backbone through ensemble voting of enhanced semantic features and further enhances the fused feature with the semantic feature of the largest receptive field.
- To reduce the significant latency caused by the final interpolation and ArgMax operation, we design the Valid Interpolate Module (VIM) that only upsamples classes present in the final prediction during inference time.

Replacing the final interpolation and ArgMax operation with VIM significantly reduces model latency.

- We combine the above modules to create a family of SOTA mobile segmentation models called PP-MobileSeg. Our extensive experiments show that PP-MobileSeg achieves an excellent balance between latency, model size, and accuracy across ADE20K and Cityscapes datasets.

2. Related Work

Under the speed and model size constraints, mobile semantic segmentation is the task that aims to adapt semantic segmentation networks with efficient network designs.

2.1. semantic segmentation

To achieve high performance in semantic segmentation, several key elements are essential, including a large receptive field to capture context [3, 36], a large resolution of features for accurate segmentation [30, 32], fusion of detail and semantic features for precise predictions [1, 22], and attention mechanisms for improving feature representation [14, 26, 31]. State-of-the-art models often combine several or even all of these elements to achieve superior performance. The primary requirement for the semantic segmentation task is that the network must be able to capture a holistic view of the scene while simultaneously preserving the image’s details and semantics. Thus, it is essential to design network architectures that can efficiently and effectively integrate these elements.

2.2. Efficient Network Designs

There are two types of efficient network architectures in the field of deep learning. The first type focuses on adding new elements to the network without introducing unwanted latency during inference. The representative one is structural reparameterization [8, 9], which approximates the multi-branch neural network block with a single branch at inference time. The second type aims to downscale the network at the expense of the model performance reduction. Designs belonging to this category include group convolution [11], channel shuffle [34], and efficient attention mechanisms [23, 28, 29].

2.3. Mobile semantic segmentation

Due to the large computational complexity of semantic segmentation, there has been limited research of segmentation on mobile devices, with only a few works focusing on this area [12, 27, 33]. Among them, TopFormer enhances the token pyramid with a self-attention block and fuses it with the local feature using their proposed injection module. Further, SeaFormer boosts the model performance with an efficient SEA attention module. Both of them significantly

outperform MobileSeg and LRASPP, which currently represent the state-of-the-art in mobile semantic segmentation.

3. Architecture

This section presents a comprehensive exploration of mobile segmentation networks designed under speed and size constraints, aiming at achieving better segmentation accuracy. Through our research, we have identified three key modules that lead to faster inference speed or smaller model size with slight performance improvements. The full architecture of PP-MobileSeg is shown in Fig. 2, which comprises four main parts: StrideFormer, Aggregate Attention Module (AAM), segmentation head, and Valid Interpolate Module (VIM). The StrideFormer takes input images and generates a feature pyramid with strided attention applied to the last two stages to incorporate global semantics. The AAM is responsible for fusing local and semantic features, which are then passed through the segmentation head to produce the segmentation mask. Finally, the upsample module VIM is used to further enhance the segmentation mask, reducing latency by only upsampling the few channels corresponding to the classes that exist in the final prediction. The following sections provide a detailed description of each of these modules.

3.1. StrideFormer

In the StrideFormer module, we utilize a stack of MobileNetV3 [12] blocks to extract features of different receptive fields. More detailed information about the variants of this architecture can be found in subsection 3.4. Given an image of $I \in R^{3 \times H \times W}$, where 3, H , W represent channels, height, and width of the image. StrideFormer produces features $\{F_{\times 8}, F_{\times 16}, F_{\times 32}\}$, representing features that are downsampled 8, 16, and 32 times compared to the resolution of the input image. One key design choice is the number of stages in the backbone, where each stage is a stack of mobilenetv3 blocks that produce one of the feature sets, $F_{\times \text{downsample-rate}}$. Inspired by efficientFormer [13], we discover that the four-stage model has minimal parameter overhead while still maintaining excellent performance compared to the five-stage model as shown in Tab. 3. Therefore we design StrideFormer with the four-stage paradigm. With $\{F_{\times 8}, F_{\times 16}, F_{\times 32}\}$ generated from the four-stage backbone, we add the M/N SEA attention blocks on the features from the last two stages following [27]. Due to the time complexity of the self-attention module with large resolution input, we add the stride convolution prior to the SEA attention module and upsample the feature afterward. In this way, we reduce computation complexity by 1/4 of the original implementation when we empower the features with global information.

3.2. Aggregated Attention Module

With the generated $\{F_{\times 8}, F_{\times 16}, F_{\times 32}\}$ from backbone, we designed an aggregated attention module (AAM) to fuse features. The structure of AAM is on the top-right of Fig. 2. Among the generated features, $\{F_{\times 16}, F_{\times 32}\}$ have a larger receptive field and contain rich semantic information. Therefore we use them as the information filter through ensemble voting to find out the important information in detail feature $F_{\times 8}$. In the filtration process, $F_{\times 16}$ and $F_{\times 32}$ are upsampled to the same resolution as $F_{\times 8}$. And sigmoid operator is applied to them to obtain weight coefficients. Afterward, $F_{\times 16}$ and $F_{\times 32}$ is multiplied and the multiplication result is used to filter $F_{\times 8}$. We can formulate the above procedure as Eq. 1

Additionally, we observed that the features with rich semantics complement the previously filtered detail feature and are crucial in improving model performance. Therefore, it should be kept to the most extent. So we add $F_{\times 32}$, the feature of the largest receptive field and enhanced with the global view, to the filtered detail feature.

$$F_{fused} = \frac{Act(F_{\times 32}) \times Act(F_{\times 16}) \times Conv(F_{\times 8})}{+ Conv(F_{\times 32})} \quad (1)$$

After feature fusion, the fused feature captures both rich spatial and semantic information, which is fundamental for segmentation performance. On top of that, we add a simple segmentation head following TopFormer [33]. The segmentation head consists of a 1×1 layer, which helps to exchange information along the channel dimension. Then a dropout layer and convolutional layer are applied to produce the downsampled segmentation map.

3.3. Valid Interpolate Module

Under the latency constraints, we did a latency profile and find out that the final interpolation and ArgMax operation take up more than 50% of the overall latency. Therefore, we designed the Valid Interpolate Module (VIM) to replace the interpolation and ArgMax operation and greatly reduce the model latency. The latency profile of the SeaFormer-Base and PP-MobileSeg-Base is shown in Fig 3. Detailed statistics after adding VIM can be seen in Table. 3.

The VIM is based on the observation that the number of classes that appear in the prediction of a well-trained model is often much less than the overall number of classes in the dataset, especially for datasets with a large number of classes. This is a common case for datasets with a large number of classes. Therefore, it is not necessary to consider all of the classes in the interpolation and ArgMax process. The structure of VIM is on the bottom-right of Fig. 2. As the structure shows, the VIM consists of three main steps. First, the ArgMax and Unique operations are applied to the down-

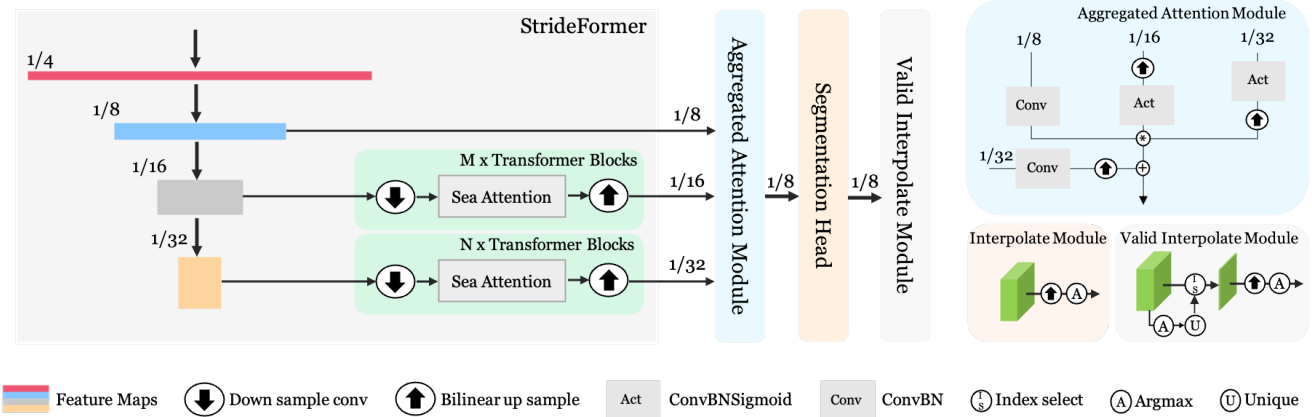


Figure 2. The architecture of PP-MobileSeg network. The structure of AAM is on the top right of the figure. The difference between the normal interpolation module and VIM is displayed at the bottom right of the figure. By selecting only the classes that exist in the final prediction, VIM significantly reduces latency by upsampling a few channels.

Method	Backbone	mIoU(%)	Latency(ms)	Parameters (M)
LR-ASPP [12]	MobileNetV3-large-x1	33.10	730.9	3.20
MobileSeg [15]	MobileNetV3-large-x1	33.26	391.5	2.85
TopFormer-Tiny [33]	TopTransFormer-Tiny	32.46	490.3	1.41
SeaFormer-Tiny [27]	SeaFormer-Tiny	35.00	459.0	1.61
PP-MobileSeg-Tiny	StrideFormer-Tiny	36.39	215.3	1.44
TopFormer-Base [33]	TopTransformer-Base	37.80	480.6	5.13
SeaFormer-Base [27]	Seaformer-Base	40.20	465.4	8.64
PP-MobileSeg-Base	StrideFormer-Tiny	41.57	265.5	5.71

Table 1. Results on ADE20K validation set. Latency is measured with PaddleLite with the final ArgMax operator on Qualcomm Snapdragon 855 CPU) and 512x512 as the input shape. All result is evaluated with a single thread. The mIoU is reported with single-scale inference.

sampling segmentation map to find out the necessary channels. Then, the index select operation selects only those valid channels, and interpolation is applied to the slimmed feature. Finally, the selected channels are upsampled to the original resolution to produce the final segmentation map. With VIM in replace of interpolation and ArgMax operation, we retrieved the final segmentation map at much less latency costs.

The use of VIM greatly reduces the channels involved in the interpolation and ArgMax operation, leading to a significant decrease in the model latency. However, VIM is only applicable when the number of classes is large enough to have channel redundancy in the model. Therefore, a class threshold of 30 is set, and VIM will not take effect when the number of classes is below this threshold.

3.4. Architecture Variants

We provide two variants of PP-MobileSeg to adapt our model to different complexity requirements, i.e., PP-

MobileSeg-Base and PP-MobileSeg-Tiny. The size and latency of these two variants with the input of shape 512x512 are shown in Tab. 1. The base and tiny model have the same number of MobileNetV3 layers, whereas the base model is wider than the tiny model. And the base model generates features with more channels to enrich the feature representation. Besides, there are several differences in the attention block as well. PP-MobileSeg-Base model has 8 heads in the SEA attention module, $M/N = 3/3$ attention blocks. The PP-MobileSeg-Tiny model has 4 heads in the SEA attention module and the number of blocks is $M/N = 2/2$. The feature channels of the last two stages are 128, 192 for PP-MobileSeg-Base and 64, 128 for PP-MobileSeg-Tiny respectively. The setting of the feature fusion module is the same for the base and tiny model and the embed channel dim of AAM is set to 256. For more details about the network architecture, please refer to the source code.

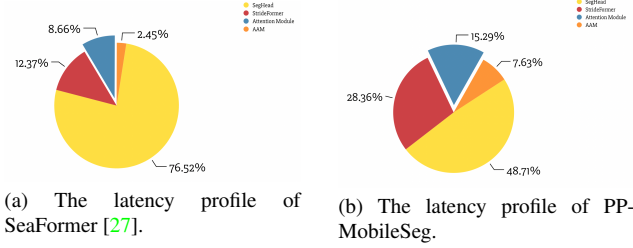


Figure 3. Latency profile compare between SeaFormer and PP-MobileSeg.

4. Experiments

In this section, we first present the dataset used for our model training and evaluation and provide implementation details on our training and inference implementation. Secondly, We compared the proposed method with the previous state-of-the-art on this task in terms of accuracy, inference speed, and model size. Finally, we perform an ablation study to demonstrate the effectiveness of our proposed method.

4.1. Experiments Setup

4.1.1 Datasets

We perform our experiments on ADE20K [36] and cityscapes [6] datasets, and the mean of class-wise intersection over union (mIoU) is used as the evaluation metric. **ADE20K** is a parsing dataset that contains 25K images in total and 150 fine-grained semantic concepts. All images are split into 20K/2K/3K for training, validation, and testing. **Cityscapes** is a large-scale dataset for semantic segmentation. It consists of 5,000 fine annotated images, 2975 for training, 500 for validation, and 1525 for test-dev. The resolution of the images is 2048×1024 , which poses a great challenge for models used in mobile devices.

4.1.2 Implementation Details

Our implementation is built upon PaddleSeg [15] and PaddlePaddle [19].

Training Settings Our backbone is pre-trained on ImageNet1K [7] to retrieve common knowledge about images. We set the batch size as 32 and train the model for 80K iterations. During training, we cross entropy loss and Lovasz loss with the loss ratio of 4:1 [2] to optimize the model. We use the exponential moving average method to average model parameters from different training iterations and the moving average coefficient is 0.999 [25]. The learning rate is at 0.006 and uses the ADAMW [17] optimizer with the weight decay set at 0.01. the learning rate schedule is set as the combination of the warmup schedule and the poly schedule with factor 1.0. The learning rate goes

up from $1e-6$ for 1500 iters and then decreases linearly. For ADE20K, we follow the data augmentation strategy of TopFormer and SeaFormer [27, 33], including the random scale ranges in $[0.5, 2.0]$, image crop to the given size, random horizontal flip, and random distortion. For Cityscapes, the data augmentation is the same except that we crop the image to 1024×512 rather than 512×512 in ADE20K datasets, and the random scale ranges in $[0.125, 1.5]$. Our model is trained with two Tesla V100 GPUs. We report the single scale results on the validation set to compare with other methods.

Inference Settings During inference, we set the input shape as 512×512 for ADE20K datasets and 512×1024 for cityscapes. To test the model latency, the full-precision PP-MobileSeg models are exported to the static model, and the latency is then measured on the Qualcomm 855 with PaddleLite on a single thread. During inference, we use VIM in place of the interpolation and ArgMax operation. It is worth noting that the image preprocesses, including resizing and normalizing, is accomplished before the inference process, so the inference time only includes model infer time. Especially, the latency of VIM is correlated with the number of classes predicted in the image. Therefore, we use an image from the ADE20K validation set, which has the average number of categories, to evaluate the latency for a reasonable comparison.

4.2. Comparison with State-of-the-arts

ADE20K Results Table. 1 presents the comparison of PP-MobileSeg with previous mobile semantic segmentation models, including both lightweight vision transformers and efficient CNNs, and report the results on parameters, latency, and mIoU. As the results show, PP-MobileSeg outperforms these SOTA models not only on latency but also on model size while maintaining a competitive edge in accuracy. Compare with MobileSeg and LRASPP, both of them use MobileNetV3 as their backbone, PP-MobileSeg-Tiny is more than 3.0 higher than them in mIoU, while being 49.47% smaller and 45% faster than MobileSeg. And PP-MobileSeg-Tiny is 55% smaller and 70.5% faster than LRASPP. In comparison with the SOTA vision transformer-based models TopFormer and SeaFormer, which use convolution-based global self-attention as their semantics extractor, PP-MobileSeg achieves higher segmentation accuracy with lower latency and smaller model size. PP-MobileSeg-Base is about the same size or 34.9% smaller and 42.9% to 44.7% faster than its counterparts, while maintaining a competitive edge in accuracy and is 1.37% to 3.77% higher in mIoU. These results demonstrate the effectiveness of PP-MobileSeg in improving feature representation.

Cityscapes Results It can be seen from Table. 2 that PP-MobileSeg-Tiny achieves better performance in all as-

Method	Backbone	mIoU(%)	Latency(ms)	Parameters (M)
SeaFormer-Small [27]	SeaFormer-Small	70.70	204.9	1.61
PP-MobileSeg-Tiny	StrideFormer-Tiny	70.82	158.3	1.44
SeaFormer-Base [27]	SeaFormer-Base	72.20	297.3	8.64
PP-MobileSeg-Base	StrideFormer-Base	74.14	323.7	5.71

Table 2. **Results on Cityscapes validation set.** We measure latency using PaddleLite with the final ArgMax operator on an ARM-based device with a single Qualcomm Snapdragon 855 CPU, with an input shape of 1024x512. As the number of classes in the Cityscapes dataset is small, we did not use the Valid Interpolate Module (VIM). All results are evaluated using a single thread, and we report the mean Intersection over Union (mIoU) using single-scale inference.

VIM	StrideFormer	AAM	mIoU	Latency(ms)	Params(M)
			40.20	465.6	8.27
✓			40.20	234.6	8.17
✓	✓		40.98	235.1	5.54
✓	✓	✓	41.57	265.5	5.71

Table 3. Ablation Study of three proposed modules of PP-MobileSeg-Base on ADE20K dataset.

pects of accuracy, latency, and parameters than SeaFormer-small. Furthermore, PP-MobileSeg-Base achieves significantly better accuracy with comparable latency and smaller model sizes. These results demonstrate that PP-MobileSeg maintains its excellent balance among accuracy, model size, and speed even under high-resolution inputs.

4.3. Ablation Study

We conduct an ablation study to discuss the influence of our proposed modules and dissect and analyze these modules. In Table 3, we show the effectiveness of three proposed modules by adding them to the baseline one by one.

VIM: As we mentioned before, VIM serves as the replacement for interpolation and ArgMax operations to accelerate the inference speed. As we can see from the profile comparison (Fig. 3), the overall latency of Segmentation greatly decreased from 76.32% to 48.71% with the application of VIM. And the experimental results from Table 3 show the model latency is decreased by 49.5% after adding VIM. These experiments prove that VIM’s acceleration capabilities on datasets with a large number of classes are exceptional.

StrideFormer: The usage of the four-stage network in StrideFormer resulted in a notable reduction of 32.19% in parameter overhead. The experimental results also show an increase in accuracy by 0.78%, which we attribute to the enhanced backbone.

AAM: AAM raises the accuracy by 0.59% while slightly increasing the latency and model size. To gain insight into the design of the AAM, we split the fusion module into two branches: the ensemble vote and the final semantics

ensemble vote	final semantics	mIoU	Params(M)
✓		41.12	5.71
	✓	41.53	5.63
✓	✓	41.57	5.71

Table 4. Ablation Study of the AAM module. Ensemble vote represents the multiplication of $F_{\times 32}$ and $F_{\times 16}$ to filter $F_{\times 8}$, without it, $F_{\times 16}$ and $F_{\times 32}$ will filter $F_{\times 8}$ the one by one as SeaFormer [27]. Final semantics represent whether we add $F_{\times 32}$ to the fused feature.

as shown in Table 4. And the reported results reveal the significance of both branches and especially the importance of final semantics. Without it, the accuracy can drop by 0.45%.

5. Conclusion

In this paper, we investigate the design options for hybrid vision backbones and addressed the latency bottleneck in mobile semantic segmentation networks. After thorough exploration, we identified the mobile-friendly design choices and propose a new family of mobile semantic segmentation networks called PP-MobileSeg with the combination of transformer blocks and CNN. With the carefully designed backbone, fusion module and interpolate module, PP-MobileSeg achieves a SOTA balance between model size, speed, and accuracy on ARM-based devices.

References

- [1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 2
- [2] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4413–4421, 2018. 5
- [3] 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. 2
- [4] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobileformer: Bridging mobilenet and transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5270–5279, 2022. 1
- [5] 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
- [6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2, 5
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5
- [8] Xiaohan Ding, Honghao Chen, Xiangyu Zhang, Kaiqi Huang, Jungong Han, and Guiguang Ding. Re-parameterizing your optimizers rather than architectures. *arXiv preprint arXiv:2205.15242*, 2022. 2
- [9] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021. 2
- [10] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chun-jing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022. 1
- [11] 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. 2
- [12] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 1, 2, 3, 4
- [13] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022. 2, 3
- [14] Huajun Liu, Fuqiang Liu, Xinyi Fan, and Dong Huang. Polarized self-attention: Towards high-quality pixel-wise regression. *arXiv preprint arXiv:2107.00782*, 2021. 2
- [15] Yi Liu, Luta Chu, Guowei Chen, Zewu Wu, Zeyu Chen, Baohua Lai, and Yuying Hao. Paddleseg: A high-efficient development toolkit for image segmentation, 2021. 4, 5
- [16] 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
- [17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5
- [18] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007. 1
- [19] Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang. Paddlepaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Computing*, 1(1):105–115, 2019. 5
- [20] Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 3–20. Springer, 2023. 1
- [21] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 1
- [22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18, pages 234–241. Springer, 2015. 2
- [23] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3531–3539, 2021. 2
- [24] 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
- [25] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 5
- [26] 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. 2
- [27] Qiang Wan, Zilong Huang, Jiachen Lu, Gang Yu, and Li Zhang. Seaformer: Squeeze-enhanced axial transformer for mobile semantic segmentation. *arXiv preprint arXiv:2301.13156*, 2023. 1, 2, 3, 4, 5, 6
- [28] 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

- [29] Jian Wang, Chenhui Gou, Qiman Wu, Haocheng Feng, Junyu Han, Errui Ding, and Jingdong Wang. Rtformer: Efficient design for real-time semantic segmentation with transformer. *arXiv preprint arXiv:2210.07124*, 2022. 2
- [30] 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. 2
- [31] 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. 1, 2
- [32] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution transformer for dense prediction. *arXiv preprint arXiv:2110.09408*, 2021. 2
- [33] Wenqiang Zhang, Zilong Huang, Guozhong Luo, Tao Chen, Xinggang Wang, Wenyu Liu, Gang Yu, and Chunhua Shen. Topformer: Token pyramid transformer for mobile semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12083–12093, 2022. 1, 2, 3, 4, 5
- [34] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 2
- [35] 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. 1
- [36] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2, 5
- [37] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023. 1