Improving Efficient Semantic Segmentation Networks by Enhancing Multi-scale Feature Representation via Resolution Path Based Knowledge Distillation and Pixel Shuffle

Biao Yang(✉), Fanghui Xue, Yingyong Qi, and Jack Xin

Department of Mathematics, University of California, Irvine, CA 92697, USA
biaoy1@uci.edu

Abstract. Multi-resolution paths and multi-scale feature representation are key elements of semantic segmentation networks. We develop two techniques for efficient networks based on the recent FasterSeg network architecture. One is to use a state-of-the-art high resolution network (e.g. HRNet) as a teacher to distill a light weight student network. Due to dissimilar structures in the teacher and student networks, distillation is not effective to be carried out directly in a standard way. To solve this problem, we introduce a tutor network with an added high resolution path to help distill a student network which improves FasterSeg student while maintaining its parameter/FLOPs counts. The other finding is to replace standard bilinear interpolation in the upscaling module of FasterSeg student net by a depth-wise separable convolution and a Pixel Shuffle module which leads to 1.9% (1.4%) mIoU improvements on low (high) input image sizes without increasing model size. A combination of these techniques will be pursued in future works.

Keywords: Multi-resolution paths · Distillation · Pixel shuffle

1 Introduction

Semantic segmentation is concerned with pixel-wise classification of images and has been studied as a long-standing problem in computer vision, see [7,14] and references therein. Predictions are first made at a range of scales, and are then combined with averaging/pooling or an attention layer. A class of efficient networks (called FasterSeg) have been recently constructed [2] based on differentiable neural architecture search [8] of a supernet and a subsequent knowledge distillation [5] to generate a smaller student net with 3.4M parameters and 27G FLOPs on full resolution (1024×2048) image input[1].

[1] This model was searched on our machine based on source code from FasterSeg [2].

G. Bebis et al. (Eds.): ISVC 2021, LNCS 13018, pp. 325–336, 2021.
https://doi.org/10.1007/978-3-030-90436-4_26

We are interested in distilling such a light weight Student Net from a high performance Teacher Net which we choose as HRNet-OCR [17] in this work. Our motivation is to improve the FasterSeg Student Net while maintaining its size and FLOPs by enhancing the resolutions of its multi-scale feature maps and their combinations for better prediction. The HRNet has about 10% higher accuracy than the FasterSeg Teacher Net on Cityscapes dataset [4]. Specifically, we first add a higher resolution path to the FasterSeg Student Net architecture and train it through distilling HRNet predictions. Then we let this high resolution path guide the prediction of the lower resolution paths in FasterSeg Student Net through a feature affinity (FA) matrix. At inference, the high resolution path is absent hence its role is virtual and does not add computational overheads on the Student Net. Though knowledge distillation at intermediate level was known in FitNets [10], the knowledge passing across multi-resolution paths for semantic segmentation appears new. In addition, we improve the inaccurate interpolation treatment in FasterSeg's feature fusion module by a combination of depth-wise separable convolutions and Pixel Shuffle (PS) technique [11], resembling the efficient operations in Shufflenets [9,18] for regular image classification task.

Our main contributions are:

1) introducing a novel *teacher-tutor-student framework* to enhance multi-resolution paths by path-wise knowledge distillation with application to Faster-Seg Student Net while keeping computational costs invariant, which utilizes the intermediate feature information from the tutor model;
2) improving multi-scale feature map fusion by depth-wise separable convolution and Pixel Shuffle techniques to gain 1.9% (1.4%) validation accuracy in mIoU on low (high) resolution input images from Cityscapes dataset, at reduced computational costs.

The rest of the paper is organized as follows. Sect. 2 is a summary of related works. Sect. 3 presents our teacher-tutor-student distillation framework. Sect. 4 shows improved semantic segmentation results by our student network with virtual high resolution path on Cityscapes data sets, and their analysis. Sect. 5 describes the Pixel Shuffle technique for multi-scale feature fusion and supporting experimental results. The concluding remarks are in Sect. 6.

2 Related Works

2.1 Overview

Semantic segmentation has been studied for decades. Recent lines of research include hierarchical architecture search, knowledge distillation (introduced in [5] for standard classification), and two-stream methods. Among large capacity models are Autodeeplabs ([7] and references therein), high resolution net (HRNet [17]), zigzag net [6] and hierarchical multi-scale attention network [14]. Among the light weight models are FasterSeg [2], and BiSenet [16]. In Gated-SCNN [13], a high level stream on region masks guides the low level stream on shape features

for better segmentation. A gated structure connects the intermediate layers of the two streams, and resulted in 2% mask (mIoU) gain over DeepLabV3+ [1] on Cityscapes dataset [4]. In [15], a dual super-resolution learning framework is introduced to produce high resolution representation on low resolution input. A 2% gain in mIoU over various baseline models is accomplished on Cityscapes data. A feature affinity function is used to promote cooperation of the two super-resolution networks, one on semantic segmentation, the other on single image super-resolution.

Though knowledge distillation at an intermediate level [10] has been known conceptually, how to set it up in the multi-resolution paths for semantic segmentation networks is not much studied. In part, a choice of corresponding locations in the Teacher Net and Student Net depends on network architecture. This is what we set out to do on FasterSeg Student Net.

Besides the conventional upscaling methods like bilinear interpolation, Pixel Shuffle (PS) is widely adopted in various multi-resolution image processing tasks. In the super-resolution task [11], an artful PS operator has been applied to the output of the convolutional layer in the low resolution, and hence has reduced the computational complexity. This technique is inherited by [15] for the super-resolution semantic segmentation, boosting the performance of the model with low resolution input.

2.2 Search and Training in FasterSeg

The FasterSeg search space [2] consists of multi-resolution branches with searchable down-sampling-path from high resolution to low resolution, and searchable operations in the cells (layers) of the branches. Each cell (layer) contains 5 operations: skip connect, 3×3 Conv, 3×3 Conv $\times2$, Zoomed 3×3 Conv and Zoomed 3×3 Conv $\times2$. The "Zoomed Conv" contains bilinear down-sampling, 3×3 Conv and bi-linear up-sampling.

Below we recall FasterSeg's architecture search and training procedure [2] in order to introduce our proposed path-wise distillation. In the search stage, the overall optimization objective is:

$$L = L_{seg}(M) + \lambda\, Lat(M) \tag{1}$$

where $Lat(M)$ is the latency loss of the supernet M; L_{seg} is the supernet loss containing cross-entropy of logits and targets from different branches. Note that the branches come from resolutions: 1/8, 1/16, 1/32, 1/8+1/32 and 1/16+1/32, as well as losses from different expansion ratios (max, min, random and architecture parameter ratio).

The architecture parameters (α, β, γ) are in a differentiable computation graph, and optimized by gradient descent. The α is for operations in each cell, β for down-sampling weight, and γ for expansion ratio. Following [8], the training dataset is randomly half-split into two disjoint sets Train-1 and Train-2. Then the search follows the first order DARTS [8]:

1) Update network weights W on Train-1 by gradient: $\nabla_w L_{seg}(M|W, \alpha, \beta, \gamma)$.

2) Update architecture α, β, γ on Train-2 by gradient:

$$\nabla_{\alpha,\beta,\gamma} L_{seg}(M|W,\alpha,\beta,\gamma) + \lambda \cdot \nabla_{\alpha,\beta,\gamma} LAT(M|W,\alpha,\beta,\gamma).$$

For teacher-student co-searching with two sets of architectures (α_T, β_T) and $(\alpha_S, \beta_S, \gamma_S)$, the first order DARTS [8] becomes:

1) Update network weights W by $\nabla_w L_{seg}(M|W,\alpha_T,\beta_T)$ on Train-1,
2) Update network weights W by $\nabla_w L_{seg}(M|W,\alpha_S,\beta_S,\gamma_S)$ on Train-1,
3) Update architecture α_T, β_T by $\nabla_{\alpha,\beta,\gamma} L_{seg}(M|W,\alpha_T,\beta_T)$ on Train-2,
4) Update architecture α_S, β_S, γ_S by $\nabla_{\alpha,\beta,\gamma} L_{seg}(M|W,\alpha_S,\beta_S,\gamma_S)$ $+\lambda\nabla_{\alpha,\beta,\gamma} LAT(M|W,\alpha_S,\beta_S,\gamma_S)$ on Train-2.

The next step is to train the weights to obtain final models.

Step 1): train Teacher Net by cross-entropy to compute the losses between logits of different resolutions and targets.

$$loss_T = CE(pred8_T, target) + \lambda\, CE(pred16_T, target) + \lambda\, CE(pred32_T, target) \tag{2}$$

where $predn_T$ is the prediction of the $1/n$ resolution path of the Teacher Net.

Step 2): train Student Net with an extra distillation loss between logits of resolution 1/8 from Teacher Net and logits of resolution 1/8 from Student Net.

$$\begin{aligned} Loss_S = {} & CE(pred8_S, target) + \lambda\, CE(pred16_S, target) \\ & + \lambda\, CE(pred32_S, target) + KL(pred8_T, pred8_S). \end{aligned} \tag{3}$$

Here $predn_S$ is the prediction of $1/n$ resolution path of the Student Net.

3 Resolution Path Based Distillation

In Sect. 3.1, we first show how to build a FasterSeg Student Net with 1/4 resolution path. Then we introduce feature affinity loss in Sect. 3.2, with search and training details in Sect. 3.3.

3.1 Tutor Model with 1/4 Resolution Path

We build a FasterSeg Student Net ("student tutor") with 1/4 resolution path, see in Fig. 1. The 1/4 resolution path follows the 1/4 resolution stem and contains 2 basic residual 2× layers also used in other stems. The 1/4 path then merges with the 1/8 resolution path by interpolating output of the 1/8 resolution path to the 1/4 resolution size, refining and adding. The model is trained with HRNet-OCR as the Teacher Net.

In Fig. 1, the stem is a layer made up of Conv and Batch normalization and ReLU. The cell is mentioned in Sect. 2.2. And the head is to fuse outputs of different resolution paths together. In 1/4 path, its output is directly added to the output of 1/8 path. While for the other paths, the outputs are concatenated with up-sampled output of lower resolution paths, and then go through a 3×3 Conv.

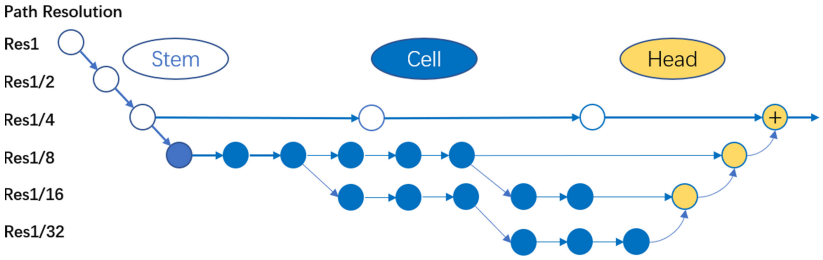


Fig. 1. Tutor net: a FasterSeg form of Student Net with 1/4 resolution path added.

3.2 Feature Affinity Loss

Feature Affinity (FA) loss [15] is a measure to overcome dimension inconsistency in comparing two feature maps by aggregating information from all channels. Consider a feature map with dimensions (H, W, C), which contains $H \times W$ vectors (in pixel direction) of length C. A pixel of a feature map F is denoted by F_i, where $1 \leq i \leq H \times W$. The affinity of two pixels is reflected in the affinity matrix with entries being the pairwise normalized inner product:

$$S_{ij} = \left(\frac{F_i}{\|F_i\|_p}\right)^T \cdot \left(\frac{F_j}{\|F_j\|_p}\right). \tag{4}$$

In other words, the affinity matrix consists of cosine similarities of all pixel-pairs. In order to make sure their consistency in the spacial dimension, we need to interpolate the student affinity matrix to the same dimensional size of the teacher affinity matrix. Let the two affinity matrices for the Teacher and Student Nets be denoted by S^t_{ij} and S^s_{ij}. Then the Feature Affinity (FA) loss is defined as [15]:

$$L_{fa} = \frac{1}{H^2W^2} \sum_{i=1}^{HW} \sum_{j=1}^{HW} \|S^t_{ij} - S^s_{ij}\|_q. \tag{5}$$

where q is not necessarily the dual of p. Here we choose $p = 2, q = 1$, and consider adding the FA loss (5) to the distillation objective for gradient descent.

Let $S^{1/n}$ be the output of the $1/n$ resolution path of Student Net after passing to ConvNorm layers, $T^{1/n}$ be the output of the $1/n$ resolution path of Teacher Net. We introduce a path-wise FA loss as:

$$FA\_loss = L_{fa}(S^{1/8}, T^{1/4}) + \lambda L_{fa}(S^{1/16}, T^{1/16}) + \lambda L_{fa}(S^{1/32}, T^{1/32}), \tag{6}$$

where λ balances the FA losses on different paths. In our experiment, $\lambda = 0.8$ is chosen to weigh a little more on the first term for the 1/8 path of the student net to mimic the 1/4 path of the tutor net. Figure 2 illustrates how our path-wise FA loss is constructed for distillation learning in the training process.

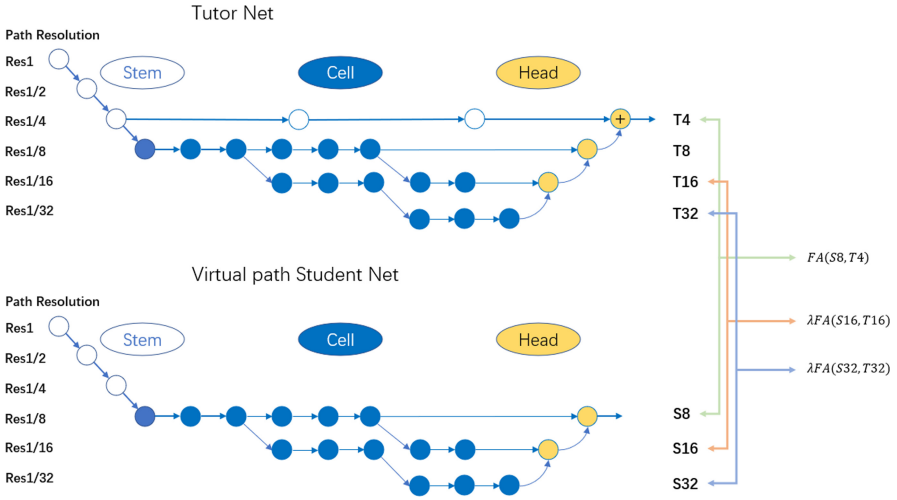


Fig. 2. Student Net with virtual 1/4 path (lower) distilled from tutor net (upper).

3.3 Teacher Net Guided Student Net Search and Training

We summarize our search and training steps below.

Search:
FasterSeg searches its own teacher model, which takes time and may not be most ideal. Instead, we opt for a state-of-the-art model as teacher to guide the search of a light weight Student Net. In our experiments, the Teacher Net is HRNet-OCR [14]. To shorten inference time, we set it back to the original HRNet [12]. The search objective is:

$$L = L_{seg}(M) + \lambda_1 \, Lat(M) + \lambda_2 \, Dist(M, HR). \tag{7}$$

The $L_{seg}(M)$ and $Lat(M)$ are same as in Eq. (1) of FasterSeg, with the added third term to narrow the distance between Student Net and the Teacher Net HR.

By first order DARTS [8] on the randomly half split training datasets (Train-1 and Train-2), we have:

1) Update network weights W by $\nabla_w L_{seg}(M|W, \alpha, \beta, \gamma)$ on Train-1
2) Update architecture α, β, γ by $\nabla_{\alpha,\beta,\gamma} L(M|W, \alpha, \beta, \gamma)$ on Train-2.

Training:

1) The baseline model is a FasterSeg student model distilled from HRNet. The training objective is:

$$\begin{aligned} Loss_S = {} & CE(pred8_S, target) + \lambda \, CE(pred16_S, target) \\ & + \lambda \, CE(pred32_S, target) + KL(pred_{HR}, pred8_S). \end{aligned} \tag{8}$$

Here $pred_{HR}$ is the prediction of the HRNet and $predn_S$ is the same notation as in FasterSeg training.

2) For the Student Net with virtual 1/4 path, we first train a FasterSeg Student Net with additional 1/4 resolution path as in Sect. 3.1. This more accurate yet also heavier temporary Student Net serves as a "tutor" for the final Student Net.

Similar to FasterSeg, we only use the outputs of the 1/4, 1/16 and 1/32 resolution paths. The output of the 1/8 path is fused with that of the 1/4 path for computational efficiency and memory savings. The resulting loss is:

$$\begin{aligned} Loss_{Tu} = & \, CE(pred4_{Tu}, target) + \lambda CE(pred16_{Tu}, target) \\ & + \lambda CE(pred32_{Tu}, target) + KL(pred_{HR}, pred4_{Tu}). \end{aligned} \tag{9}$$

Here Tu stands for the "tutor" model with 1/4 resolution path.

Next we distill the true Student Net from the "tutor net" by minimizing the loss function:

$$\begin{aligned} Loss_S = & \, cFA\_loss + CE(pred8_S, target) + \lambda CE(pred16_{Tu}, target) \\ & + \lambda CE(pred32_{Tu}, target) + KL(pred4_{Tu}, pred8_S), \end{aligned} \tag{10}$$

where $predn_{Tu}$ is the prediction of the tutor model.

4 Experiments

In Sect. 4.1, we introduce the dataset and our computing environment. We present experimental results and analysis in Sect. 4.2, and analyze FA loss in Sect. 4.3.

4.1 Dataset and Implementations

We use the Cityscapes [4] dataset for training and validation. There are 2975 images for training, 500 images for evaluation, and 1525 images for testing. The class mIoU (mean Intersection over Union) is the accuracy metric.

Our experiments are conducted on Quadro RTX 8000. Our environment is CUDA 11.2 and CUDNN 7.6.5, implemented on Pytorch.

4.2 Experimental Results and Analysis

We perform our method on both low resolution (256×512) input and high resolution (512×1024) input. The different resolution here means the cropping size of the raw image input. The evaluation is on the original image resolution of 1024×2048. And we use only the train dataset for training and perform validation on validation (Val) dataset.

During the search process, we set pretrain epochs as 20, number of epochs as 30, batch size as 6, learning rate as 0.01, weight decay as 5.e−4, initial latency weight λ_1 as 1.e−2, distillation coefficient λ_2 as 1.

We first train a baseline student model with total epochs as 500, batch size as 10, learning rate as 0.012, learning rate decay as 0.990, weight decay as 1.e−3. The baseline model is comparable to FasterSeg's student [2] in performance.

Then we train our 1/4 path tutor model for low (high) resolution input as a student initialized from the baseline model above with HRNet as teacher. The total number of epochs is 350 (400 for high), batch size is 10, learning rate is 0.0026 (0.003 for high), learning rate decay is 0.99, and weight decay is 1.e−3.

Finally, we distill the student model with virtual 1/4 path from the 1/4 path tutor model with 400 epochs, batch size 10, learning rate 0.003, learning rate decay 0.99, weight decay 1.e−3 and coefficient c for FA loss as 1.

The results are in Table 1 where the baseline Student Net has 60.1% (71.1%) mIoU for low (high) resolution input. The tutor net with 1/4 path has 63.6% (73.03%) mIoU for low (high) resolution input. The Student Net with virtual 1/4 path increases the accuracy of the baseline Student Net to 62.2% (72.3%) mIoU for low (high) resolution input. While we have trained the Student Net for different input sizes (256×512 and 512×1024), the inferences are made on the full resolution (1024×2048). For all of our experiments, FLOPs is also computed on the full resolution (1024×2048) images, regardless of the training input size. The improvement by the student net with virtual 1/4 path over the baseline student net are illustrated through images in Fig. 3. In the rectangular regions marked by dashed red lines, more pixels are correctly labeled by the student net with virtual 1/4 path. On test dataset, the baseline Student Net has 57.7% (69.3%) mIoU for low (high) resolution input, and the virtual 1/4 path Student Net has 60.2% (69.7%) mIoU for low (high) resolution input.

We show ablation experimental results for low resolution input in Table 2. It contains student nets with virtual 1/4 path trained from scratch and different coefficient c settings for the FA loss in Eq. (10). Both fine tuning and FA loss contribute to the improvement of the accuracy.

To further understand our teacher-tutor-student distillation framework, we studied direct student net distillation from HRNet with the help of FA loss. However, the mIoU is lower than that from the above teacher-tutor-student distillation framework. This might be due to the more disparate architectural structures between the teacher model and the FasterSeg form of the student net. We notice that our improvement is lower for the high resolution input. This may be due to reaching the maximal capability of the student net. In our experiment, we only have 2 layers in the 1/4 resolution path of the student net. For more gain in mIoU, adding more layers in the path is a viable approach to be explored in the future.

4.3 Analysis of FA Loss

In [15], the authors added a ConvNorm layer right after the extraction of feature maps to adjust their distributions before computing FA loss. We found that it

Table 1. Tutor/student net with virtual 1/4 path for low/high image input sizes on Cityscapes Val dataset.

Input size	256×512	512×1024	Param/FLOPs
Baseline student net	60.1	71.1	3.4M/27G
Tutor net w. 1/4-path	63.6	73.0	3.9M/100G
Student net (w. virtual 1/4-path)	62.2	72.3	3.4M/27G

Table 2. Student nets with virtual 1/4 path on low input size images of Val dataset.

Input size	256×512
Baseline student net	60.1
Student net (w. virtual 1/4-path) from scratch	59.9
Student net (w. virtual 1/4-path) $c = 0$	60.9
Student net (w. virtual 1/4-path) $c = 1$	62.2

would be better not add extra layers to the tutor model in our case as the tutor model is already fixed. By checking the affinity matrices with and without ConvNorm layers, we observed that given the tutor model, adding such extra layers to both student and tutor models before computing FA loss forces the entries of affinity matrices all close to 1 (a trivial way to reduce FA loss). This phenomenon is illustrated in Fig. 4. And we also find that enlarging the coefficient c in FA loss will help the model converge faster but might cause overfitting.

5 Pixel Shuffle Prediction Module

In this section, we develop a specific technique for FasterSeg student prediction module, see Fig. 5. FasterSeg [2] uses Feature Fusion Module (FFM) to fuse two feature maps from different branches, with the outcome of size $C \times H \times W$, which is passed through the Head Module to generate the prediction map of size $19 \times H \times W$. Afterward, a direct interpolation by a factor of 8 up-scales the prediction map to the original input size. This treatment makes FasterSeg fast however at an expense of accuracy.

We discovered an efficient improvement by generating a larger prediction map without introducing too many parameters and operations. The idea is to adopt depth-wise separable convolution [3] to replace certain convolution operations in FasterSeg. First, we reduce the channel number of FFM by half and then use Refine Module (group-wise convolution) to raise the channel number. Finally, we apply Pixel Shuffle on the feature map to give us a feature map of size $C/2 \times 2H \times 2W$. Let us estimate the parameters and operations of FFM and Heads focusing on the convolution operations. The FFM is a 1×1 convolution with C^2 parameters and C^2HW operations. The Head contains a 3×3 Conv and a 1×1 Conv, whose parameters are $9\,C^2 + C\,N$ and the operations are $9\,C^2HW + CNHW$. In

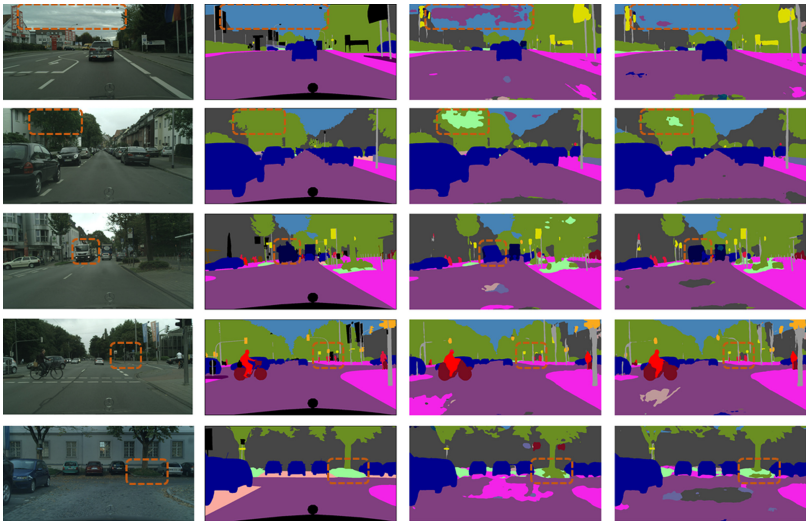


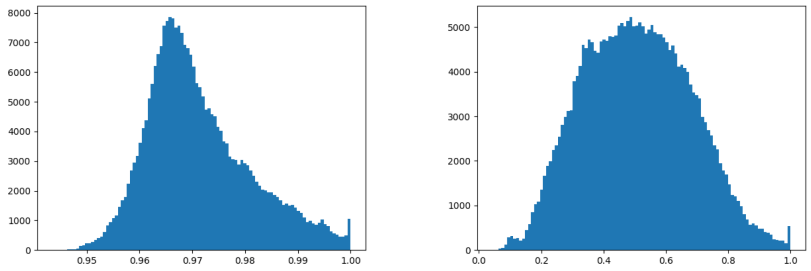
Fig. 3. Baseline student net improved by student net with virtual 1/4-path (pixels in rectangular regions). The 4 columns (left to right) display input images, true labels, output labels of the baseline net and the student net with virtual 1/4-path resp. The 5 example images are taken from the validation dataset.

Table 3. Comparison of validation mIoUs of our proposed up-scaling method with depth-wise separable convolution (dep. sep. conv) and Pixel Shuffle (PS) on low/high input image sizes vs. those of the FasterSeg student (original net) [2] implemented on our local machine.

Input size	256×512	512×1024	Param/FLOPs
Original net	60.1	71.1	3.4 MB/27 GB
Our net with dep. sep. conv & PS	62.0	72.5	3.3 MB/27 GB

total, there are $10\,C^2HW + NCHW$ and $10\,C^2 + NC$ operations. Similarly in our structure, FFM consumes $C^2/2$ parameters and $(C^2/2)HW$ operations, Refine costs $50C$ parameters and $50CHW$ operations, the Head takes $9\,(C/2)^2 + (C/2)N$ parameters and $9(C/2)^2 2H2W + (C/2)N2H2W$ operations. In total, there are $(11/4)\,C^2 + NC/2 + 50\,C$ parameters and $9.5\,C^2HW + 2\,CNHW + 50\,CHW$ operations. If C is large enough, our proposed structure has fewer parameters and operations to produce a larger prediction map.

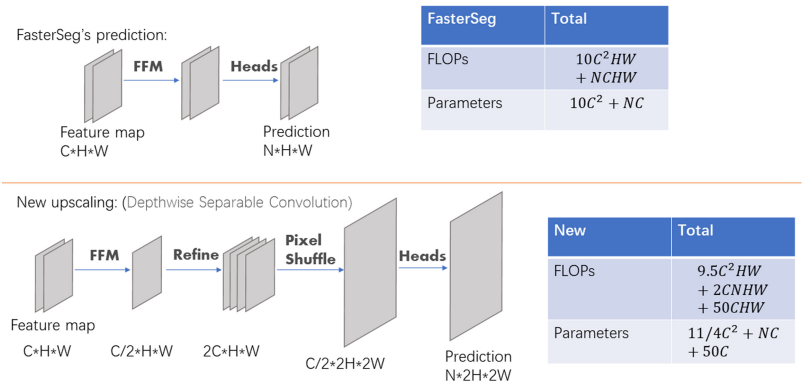
In our experiments, we replace the Prediction Module of FasterSeg model with our Pixel Shuffle Prediction Module, keeping all the other choices in training the same. As shown in Table 3, our proposed model has 0.1 MB fewer parameters, yet has achieved 1.9 % (1.4 %) mIoU improvement for low (high) resolution input images.



(a) Histogram of S_{ij} with ConvNorm layers added to feature maps prior to computing FA loss from Eq. (10).

(b) Histogram of S_{ij} from Eq. (10).

Fig. 4. Histograms of affinity matrix entries in tutor-student distillation learning.



FasterSeg	Total
FLOPs	$10C^2HW + NCHW$
Parameters	$10C^2 + NC$

New	Total
FLOPs	$9.5C^2HW + 2CNHW + 50CHW$
Parameters	$11/4C^2 + NC + 50C$

Fig. 5. Proposed pixel shuffle prediction module vs. that of FasterSeg [2].

6 Conclusion

We presented a teacher-tutor-student resolution path based knowledge distillation framework and applied it to FasterSeg Student Net [2] guided by HRnet. While preserving parameter sizes and FLOPs counts, our method improves mIoU by 2.1% (1.2%) on low (high) input image sizes on Cityscapes dataset. We designed a depth-wise separable convolution and Pixel Shuffle technique in the resolution upscaling module which improved FastserSeg's Student Net by 1.9% (1.4%) on low (high) input image sizes with slightly lower (same) parameter (FLOPs) count. In future work, we plan to add more cells on the 1/4 resolution path, combine the two methods developed here for further improvements.

Acknowledgements. The work was partially supported by NSF grants DMS-1854434, DMS-1952644, and a Qualcomm Faculty Award. The authors would like to thank Dr. Shuai Zhang and Dr. Jiancheng Lyu for helpful and enlightening discussions, and ISVC2021 reviewers for their constructive comments.

References

1. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
2. Chen, W., Gong, X., Liu, X., Zhang, Q., Li, Y., Wang, Z.: Fasterseg: Searching for faster real-time semantic segmentation. ICLR, 2020; arXiv 1912.10917 (2019)
3. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of IEEE CVPR, July 2017
4. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
5. Hinton, H., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NeurIPS-Workshop (2014)
6. Lin, D., Shen, D., Shen, S., Ji, Y., Lischinski, D., Cohen-Or, D., Huang, H.: Zigzag-net: fusing top-down/bottom-up context for object segmentation. In: CVPR (2019)
7. Liu, C., Chen, L., Schroff, F., Adam, H., Wei, H., Yuille, A., Li, F.: Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. CVPR (2019). arXiv:1901.02985v2 (2019)
8. Liu, H., Simonyan, K., Yang, Y.: DARTS: Differentiable architecture search. ICLR (2019). arXiv preprint arXiv:1806.09055 (2018)
9. Lyu, J., Zhang, S., Qi, Y., Xin, J.: Autoshufflenet: Learning permutation matrices via an exact Lipschitz continuous penalty in deep convolutional neural networks. KDD (2020)
10. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets (ICLR, 2015)
11. C Shi, W., et al.: 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, pp. 1874–1883 (2016)
12. Sun, K., et al.: High-resolution representations for labeling pixels and regions. arXiv: 1904.04514 (2019)
13. Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-SCNN: gated shape CNNs for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5229–5238 (2019)
14. Tao, A., Sapra, K., Catanzaro, B.: Hierarchical multi-scale attention for semantic segmentation. arXiv: 2005.10821 (2020)
15. Wang, L., Li, D., Zhu, Y., Tian, L., Shan, Y.: Dual super-resolution learning for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3774–3783 (2020)
16. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: BiSenet: Bilateral segmentation network for real-time semantic segmentation (ECCV, 2018)
17. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. arXiv preprint arXiv:1909.11065v5 (ECCV, 2020)
18. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: CVPR (2017)