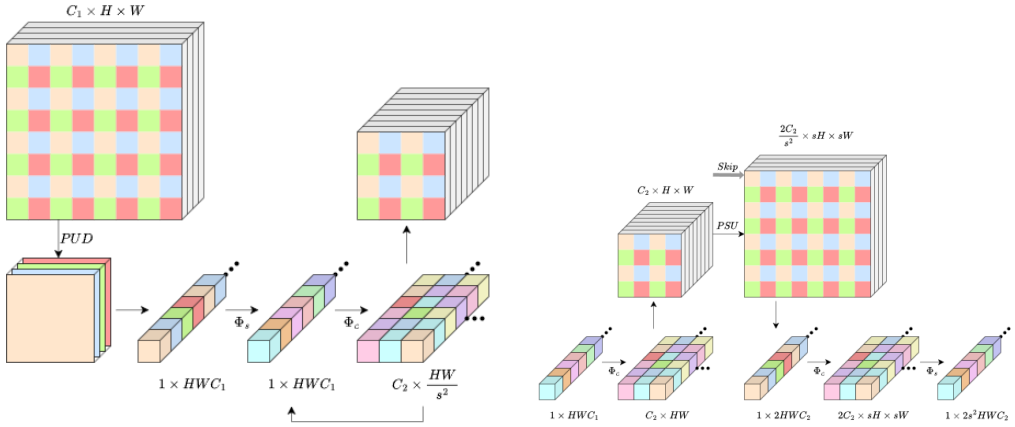
OneNet: A Channel-Wise 1D Convolutional U-Net for Unified Image Classification and Segmentation

Sanghyun Byun byuns@usc.edu
Kayvan Shah kpshah@usc.edu
Ayushi Gang agang@usc.edu
Christopher Apton apton@usc.edu

Abstract

Many modern computer vision architectures rely on U-Net for its adaptability and efficient feature extraction capabilities. However, the multi-resolution convolutional structure of these models often makes them computationally heavy, limiting their applicability on edge devices. In this paper, we demonstrate that U-Net architecture can be simplified to a 1D convolutional architecture without significant loss in accuracy, thus enhancing deployability. We propose a streamlined pipeline that performs semantic segmentation using only channel-wise 1D convolutions and pixelShuffle operations. PixelShuffle has been shown to improve accuracy in several state-of-the-art super-resolution methods while reducing computational demands. Similarly, OneNet incorporates pixel-unshuffle into the encoder, effectively capturing spatial relationships without needing 2D convolutions. This implementation reduces the parameter count by up to 71%. Our proposed approach is benchmarked against U-Net baselines with varying depths and complexities on multiple segmentation datasets, including PASCAL VOC, Oxford Pet, and MSD. While we primarily focus on the segmentation task for comparison, this architecture can be readily adapted to other convolutional pipelines.



(a) (b)

Figure 1: **Channel-Wise 1D Convolution Block** (a) Downstream convolution block used in encoder architecture. Pixel-unshuffle downsampling is used instead of a typical maxpooling operation, followed by a single spatial convolution and two channel-wise convolutions. (b) Upstream convolution block used in decoder architecture. Pixel-shuffle upsampling is used instead of interpolation or transposed convolution. Different from downstream blocks, spatial convolution is done after channel convolution.

Preprint. Under review.

1 Introduction

With advancements in model architecture, the accuracy and versatility of vision models have reached unprecedented levels. However, deploying these high-parameter models on edge devices such as mobile phones poses significant challenges due to their limited computational resources. Techniques like quantization and optimization are essential to enable the use of state-of-the-art models on these devices. Many recent vision models rely on the U-Net [20] architecture for image processing and generation, yet its structure is not optimized for efficiency in resource-constrained environments. Although the architecture is highly adaptable, minimal research has been conducted to streamline its size for edge deployment. In this study, we propose modifying the U-Net [20] backbone to reduce the number of parameters, thereby decreasing server costs and training times. This optimization would make U-Net [20] more feasible for edge deployment and open up possibilities for more complex models by reallocating resources to tasks of greater importance.

Many modern architectures, including diffusion models and VAEs, rely heavily on the U-Net [20] architecture as an encoder-decoder backbone. However, these approaches often overlook the architectural inefficiencies inherent in U-Net [20]. Since they typically employ a standard convolutional backbone such as ResNet [8], the parameter count can escalate rapidly, impacting efficiency.

In contrast, areas like image super-resolution have long benefited from techniques like PixelShuffle [21], which significantly streamline processing pipelines without compromising spatial information. Despite the clear advantages of these scaling techniques, they have not been widely explored in other domains. Additionally, while lightweight architectures like MobileNet [10] have been effective for more straightforward tasks such as classification, they remain underutilized in generative models. This gap suggests an opportunity to explore alternative, efficiency-driven architectures for more demanding tasks, potentially unlocking new performance levels and adaptability in model deployment.

In this paper, we propose a novel adaptation of the U-Net [20] architecture that bridges the gap between state-of-the-art performance and edge-deployability by reducing model size with a minimal impact on accuracy. Our approach is the first to leverage channel-wise 1D convolutions in conjunction with pixel-unshuffling to enable efficient feature extraction and spatial attention without relying on 2D computations. By eliminating 2D operations, which are often challenging to parallelize on resource-constrained edge devices, we ease the burden on sequential computing cores and make the model more suitable for edge deployment. Additionally, we optimize spatial processing by reducing kernel sizes, focusing instead on cross-feature interactions to further minimize memory requirements. This streamlined architecture can seamlessly replace the standard U-Net in existing pipelines, offering a versatile and high-efficiency solution for edge applications. Our major contributions are as follows:

- We design a novel convolution block that only uses 1D convolutions and retains spatial information through the introduction of pixel-unshuffling downsampling and channel-wise 1D convolutions.
- We implement two versions of U-Net [20] (1D encoder with 2D-decoder and 1D encoder-decoder) using our novel convolution block, effectively reducing the model size by 47% and 71% while maintaining reasonable accuracy.
- We evaluate our proposed method variations on 5 semantic segmentation datasets and compare our results to commonly used backbones for U-Net to display retention of performance while reducing the total size of the model and computations.

2 Related Works

2.1 1D Convolutions

Recent research by Kirchmeyer et al. [12] demonstrates that a ConvNet consisting entirely of 1D convolutions can do just as well as 2D on ImageNet classification. Building on this, our model aims to show that 1D convolutions can achieve similar performance on image segmentation tasks, making them more efficient for edge devices.

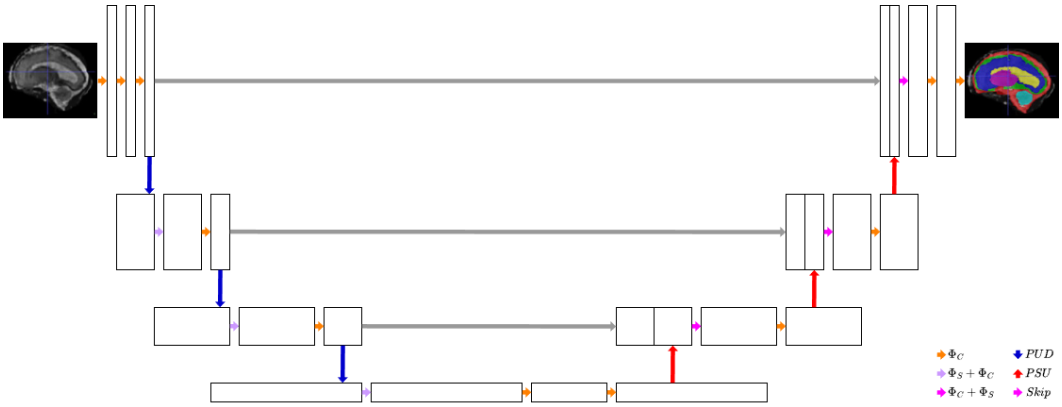


Figure 2: **Channel-Wise 1D Encoder-Decoder** OneNet employs a U-Net [20] architecture with skip connections for segmentation tasks. The encoder block replaces the maxpool with a pixel-unshuffling operation. The decoder block replaces the upsampling with a pixel-shuffling operation. Both encoder and decoders consist of a single spatial 1D convolution followed by 2 channel-wise 1D convolutions. The top layer of the decoder is implemented without batch normalization or ReLU to avoid zero-centering of the prediction head.

2.2 U-Nets

U-Nets, developed by Ronneberger et al. [20], introduced a new approach to semantic segmentation that uses a contracting path followed by an expansive path, then maps each pixel to its predicted class in the final layer. Developed to perform well on smaller datasets, U-Nets surpassed the sliding-window convolutional network [3] used prior to this. Over time, U-Nets and their variations have become a cornerstone in the medical imaging field [27, 16], especially for MRI image segmentation. For example, Gupta et al. [7] utilized U-Nets for brain tumor segmentation with notable success. Similarly, SAM-2 [19] extended this approach by introducing spatiotemporal mask predictions for videos, while Zhuang et al. [28] proposed a novel idea to tackle the challenge of lack of segmentation labels in video data by using a temporally-dependent classifier (TDC) to mimic the human-like recognition procedure. Despite these advances, however, standard U-Net architectures remain computationally demanding for edge devices. This limitation motivates our work, where we adapt the U-Net structure to reduce model size and computational load, making it more suitable for deployment on resource-constrained devices.

2.3 Fast & Accurate Segmentation

The demand for fast yet efficient segmentation models has led to the development of models like PP-MobileSeg [24], Blitzmask [2], or Mobilevig [15], which are designed specifically for mobile devices. Although PP-MobileSeg [24] achieves real-time segmentation on lightweight architectures, it struggles when dealing with large datasets that require high spatial attention. Our approach addresses this limitation by employing a state-of-the-art technique that simplifies spatial attention without sacrificing efficiency. Similarly, TFNet [13] focuses on fast and accurate segmentation for LiDAR data, which aligns with our overall aim of developing efficient, high-performance models for resource-constrained environments.

2.4 Pixel Shuffle

The concept of pixel shuffle, as proposed by Shi et al. [21], increases image resolution by converting low-resolution (LR) images into high-resolution (HR) outputs at the very end of the network making it more efficient. The model takes an $W \times H \times Cr^2$ image and converts it into a high-resolution $Wr \times Hr \times C$ image. This technique plays a crucial role in our model to upsample in the expansive path. Conversely, we efficiently apply pixel unshuffle to down-sample in the contracting path, helping capture spatial relationships while reducing computational load.

Algorithm 1 1D pixel-unshuffle downsampling

Input: Input X in shape (B, C, S), Height H, Width W

$I \leftarrow []$
for $i \leftarrow 0$ to $\frac{W}{2}$ **do**
  for $j \leftarrow 0$ to $\frac{H}{2}$ **do**
    $I \leftarrow 2i + 2j$
    $I \leftarrow 2i + 2j + 1$
    $I \leftarrow 2i + 2j + w$
    $I \leftarrow 2i + 2j + w + 1$
  end for
end for
$x \leftarrow x[:, :, I].reshape(B, C, -1, 4).transpose(0, 2, 1, 3).flatten(dim = 1)$

2.5 Diffusion Models

FreeU [22] introduces a simple method that enhances U-Net's de-noising capability at no extra cost, making it highly efficient. Their analysis shows that the backbone plays a key role in handling denoising, while the skip connections mainly bring high-frequency details to the decoder. This concept aligns with our model, which also aims to leverage diffusion models to achieve high-quality segmentation. Additionally, HarmonyView [26] explores diffusion-based sampling techniques for balancing consistency and diversity in single-image to 3D generation.

2.6 Pre-training and Fine-Tuning

Recent advancements in transformer architectures, such as Dosovitskiy et al. [5], demonstrate that pre-training on large amounts of data and then fine-tuning to multiple mid-sized or small image recognition benchmarks attains excellent results while requiring substantially fewer computational resources to train. Herzog et al. [9] explore cross-domain adaptation, emphasizing learning robust feature representations even with limited data when fine-tuning models on different datasets. This aligns with our approach, as we pre-train OneNet on a general dataset and fine-tune it for specific datasets. By adapting the model's features to new domains, we aim to achieve high performance with minimal data.

3 Methods

We employ a similar architecture to that of a UNet, except with 1D replacements and pixel-unshuffle and pixel-shuffle for downsampling and upsampling, respectively.

3.1 1D-Kernel Convolution

Similar to depth-wise convolution proposed by Howard et al. [10], we separate the task of convolutions into spatial and feature tasks.

As pixel-unshuffle downsampling transfers spatial knowledge to the channel axis, we replace spatial convolution with a 1D convolution on a flattened tensor of size (B, HWC). As it is flattened, no padding will be needed, and it would work in a similar fashion to that of reflective padding.

Channel-wise convolution processes an input tensor of size (B, HWC_1) and runs C_2 1D convolutions with a kernel size and stride of C to attain a tensor of size (B, C_2, HW). As spatial information is still intact and compared through spatial convolution, running a channel-wise convolution results in consideration of both channel-wise and spatial information.

Although our proposal works in a similar manner to depth-wise separable convolution, we inherently differ as our method can attend to spatial information as well during the second convolution layer without additional overhead.

Method	VOC [6]			PET [17]			MSD Heart [1]			MSD Brain [1]		
	$\mathcal{L}_{CE}$	$\text{mAP}_{0.5}$	mIOU	$\mathcal{L}_{CE}$	$\text{mAP}_{0.5}$	mIOU	$\mathcal{L}_{CE}$	$\text{mAP}_{0.5}$	mIOU	$\mathcal{L}_{CE}$	$\text{mAP}_{0.5}$	mIOU
U-Net_4 [20]	1.985	X	X	2.206	X	X	0.02235	X	X	0.0455	X	X
ResNet_{34} [8]	1.321	X	X	0.648	X	X	0.00874	X	X	0.1305	X	X
ResNet_{50} [8]	1.079	X	X	1.027	X	X	0.00857	X	X	0.0496	X	X
MobileNet [10]	X	X	X	X	X	X	X	X	X	X	X	X
$\text{OneNet}_{e,4}$	2.144	X	X	2.713	X	X	0.00410	X	X	0.0345	X	X
$\text{OneNet}_{ed,4}$	X	X	X	X	X	X	X	X	X	X	X	X

Table 1: **Baseline Comparisons on Semantic Segmentation Datasets** U-Net and OneNet are trained on datasets [6, 17, 1] without pre-training for a fair comparison, as outlined by the original U-Net [20]. ResNet [8] is pre-trained on ImageNet-1K for accuracy comparison. U-Net_i stands for vanilla U-Net [20] encoder with downsampling layers of i. Resnet_i stands for i-layer version of ResNet [8]. $\text{OneNet}_{e(d),i}$ has i downsampling layers, with e and d each standing for encoder and decoder, respectively.

3.2 1D Encoder

As seen in Figure 1a, a layer of the encoder used for OneNet consists of a pixel-unshuffle, 1D spatial convolution, and two channel-wise 1D convolutions.

Maxpool operation is traditionally used for downsampling in vision tasks due to its simplicity. However, the operation results in a sacrifice of spatial data that may be helpful in downstream tasks. Thus, we propose to use pixel-unshuffle in the place of max pool, which theoretically allows us to have improved spatial information retention. Further, as pixel-unshuffling folds $k \times k$ blocks into k^2 channels, it transfers spatial relationship to channel-wise relationship, allowing channel-wise 1D convolution to work similarly to a 2D spatial convolution.

Although pixel-unshuffling is beneficial, implementing 2D pixel-unshuffling would require multiple flattening and restoring calls. Thus, despite the figure displaying a 2D image basis for downsampling, we customize the design of the pixel-unshuffling operation to preserve the tensor in the 1D state to minimize tensor reshaping. We show our adaptation in Algorithm 1.

We do not test our model on a pre-trained encoder as per U-Net [20] implementation. As the paper focuses on the backbone-replacement potential of 1D convolutions, pre-training is out of the scope of our paper.

3.3 Segmentation Decoder

We perform PixelShuffle [21] as shown in 1b for upscaling. In comparison to the encoder, however, as we only upsample, the spatial information is not moved to the channel dimension. Despite this difference, if the OneNet decoder is used in conjunction, we can successfully assume that the channel dimension already consists of sufficient spatial information due to the unshuffling steps performed. This does signify that although the encoder can be used in conjunction with other decoders, the OneNet decoder is specific to the OneNet encoder.

Otherwise, the decoder follows a standard U-Net [20] architecture with skip connections. As the upsampling method reduces the channel size to one-fourth, we do not change the channel number after the concatenation of tensors.

3.4 Model Validation

We validate our approach on multiple datasets and report metrics such as mean average precision and intersection-over union. More details are discussed in section 4.

3.5 Obstacles Faced

A main obstacle faced by the proposed method is computation time. Both pixel-unshuffling and pixel-shuffling operations are expensive, and transposing and reshaping the tensor multiple times gives our model an unexpected overhead. However, we have reduced this overhead by reducing

Method	# Param (M)	Param (MB)	FLOPS (GB)	Memory (GB)
U-Net$_4$ [20]	31.04	124.03	104.72	509.61
U-Net$_5$ [20]	124.42	497.41	130.80	524.29
ResNet$_{34}$ [8]	25.05	98.07	29.40	**241.17**
ResNet$_{50}$ [8]	74.07	287.83	84.98	450.36
MobileNet [10]	X	X	X	X
OneNet$_{e,4}$	16.39	65.42	78.42	639.63
OneNet$_{e,5}$	65.73	262.63	98.82	656.41
OneNet$_{ed,4}$	**9.08**	**36.30**	**23.08**	861.93
OneNet$_{ed,5}$	36.38	145.47	39.00	885.00

Table 2: **Comparison on Model Size** Number of parameters (in millions), parameter size, FLOPS used, and memory used during inference is reported. A sample tensor of size $(1, 3, 256, 256)$ was used as the network input. The best results are shown in **bold**. U-Net$_i$ stands for vanilla U-Net [20] encoder with downsampling layers of i. Resnet$_i$ stands for i-layer version of ResNet [8]. OneNet$_{e(d),i}$ has i downsampling layers, with e and d each standing for encoder and decoder, respectively.

reshaping steps with a modified pixel-unshuffling algorithm (3.2) and hope to develop a similar method for pixel-shuffling operation.

In addition, we are in pursuit of a decoder architecture that would allow it to be universally used with other 2D encoders, similar to that of our 1D encoder.

3.6 Updated Timeline

Week 10: Model training on Imagenet-1K, COCO [14], Oxford Pet [17]

Week 11: Initial model finalization and reporting

Week 12: Further dataset training (NYUv2 [23], Diode [25]) and ablation studies

Week 13: Ablation studies

Week 14: Final Report

4 Experiments and Results

4.1 Architecture Setups

We implement OneNet architecture with PyTorch [18] and employ the Adam [11] optimizer with an initial learning rate of 1×10^{-4}, with 0.1 scale learning rate step every 20 epochs starting from epoch 50. OneNet is trained for 200 epochs with a batch size of 16 and image resolution of 256×256. We do not employ batch normalization or ReLU activation to the top decoder layer to allow segmentation mask generation. We set the initial bottle channel count to 64, the kernel size of the spatial convolution to 9, and the pixel-unshuffling operation scale to 2. All datasets are trained on a single RTX 4090 with 24GB of VRAM.

4.2 Datasets

We evaluate our method on 4 datasets: PASCAL VOC [6], Oxford Pet [17], MSD Heart [1], and MSD Brain [1]. The datasets consist of 21, 38, 4, and 4 classes, respectively, which are split into train, validation, and test sets. Following Ronneberger et al. [20], we do not pre-train the encoder for U-Net [20] and OneNet. ResNet [8] is pre-trained on Imagenet-1K [4]. Cross entropy loss, mean average precision with IOU over 0.5 (mAP), and mean intersection-over-union (mIOU) are reported for all models. The results are shown in Table 1.

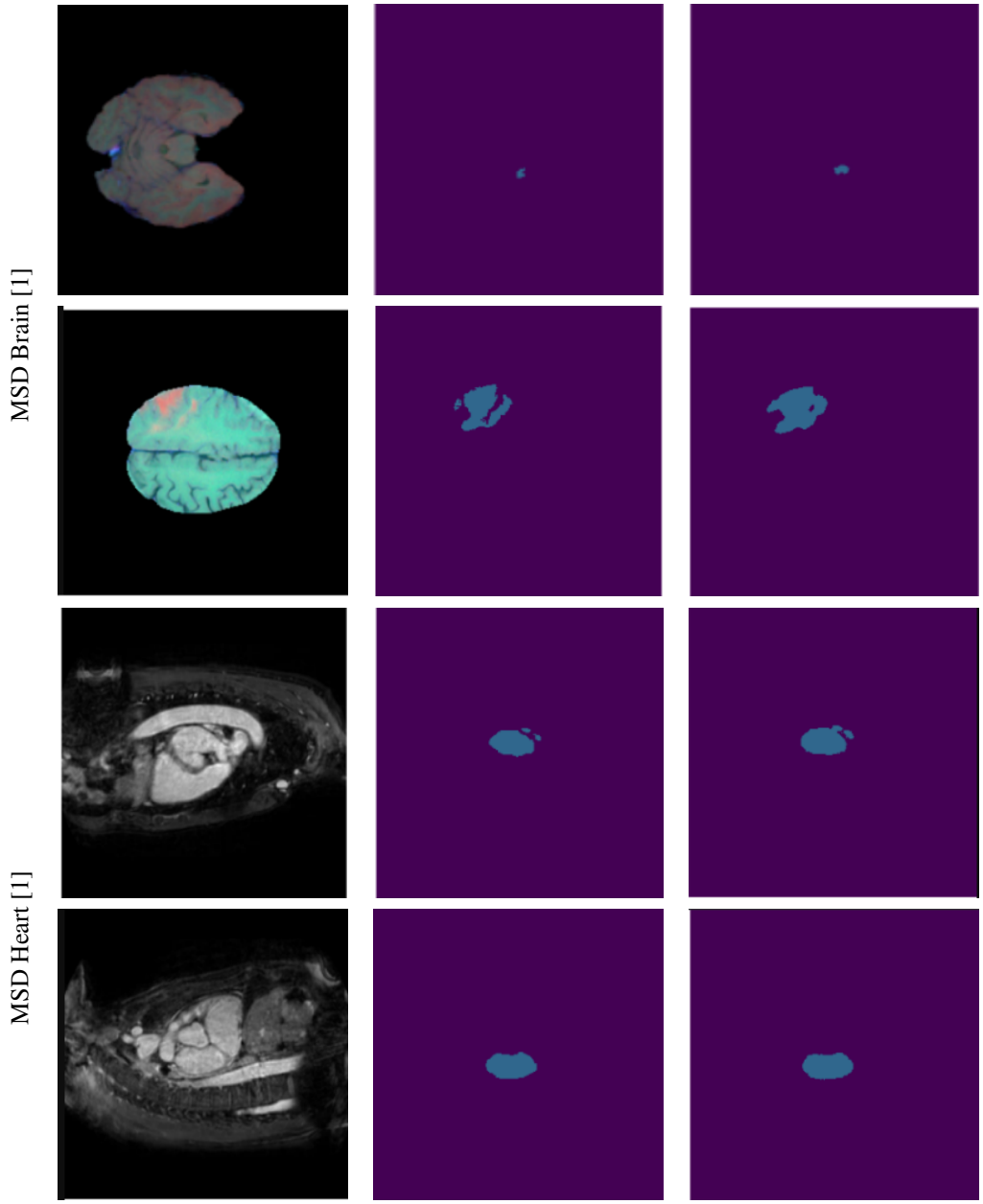


Figure 3: **Qualitative Results** Sample ground-truth and OneNet predictions on MSD Brain [1], and MSD Heart [1].

For the models with numbers shown, we have them trained but have not yet run the evaluation function. Further, we have not yet trained the models on COCO [14], NYUv2 [23], and Imagenet-1K [4] datasets. If time allows, we will pre-train all backbones with Imagenet-1K [4] to further prove its efficiency.

5 Discussion

5.1 Discussion on Results

We discuss the accuracy of OneNet in Table 1. We set the baseline as U-Net [20] for a fair comparison. Although we see a slight drop in accuracy for VOC [6], and Pet [6], it is on par with results attained

from the 2D backbone. For MSD datasets [1], we even see an improvement over 2D models. This mostly agrees with the hypothesis that there will be a slight trade-off between the accuracy and model size, except for the improvement. We hypothesize the retention of spatial information through pixel unshuffle is the cause.

However, as seen in the table, some values are missing (marked with X). Although the models have been trained, we could not fill the values in time for the report. As these values simply require evaluation. They are not included in the timeline. As discussed, we will further test the model on diverse datasets to make a better conclusion.

5.2 Discussion on Model

In Table 2, we report the parameter count and FLOPS of the proposed OneNet in comparison to the baselines [20, 8, 10] using a sample input tensor of size $(1, 3, 256, 256)$. We see roughly a 71% reduction in model parameter size and a 78% reduction in calculations, which agree with the original hypothesis. However, we see an increase in memory usage due to under-optimized PixelShuffle [21] operations, as we constantly call reshaping and axis-reordering methods.

As we improve the model for better accuracy, this number may change accordingly, depending on the assigned hyperparameter values.

References

[1] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A. Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M. Summers, Bram van Ginneken, Johanna Kirchberg, Fabian Isensee, Klaus H. Maier-Hein, M. Jorge Cardoso, Ruben Janssens, Paul F. Jäger, Simon Kohl, Laura Lange, Suprosanna Shit, Christian Siegel, Patrick Wagner, Allan Hanbury, Hans-Alois Hofmann, Joan Ruiz-Espana Tirindelli, Valentina Venturini, Benjamin Walter, and Wolfgang Brauer. The medical segmentation decathlon, 2022.

[2] Vitalii Bulygin, Dmytro Mykheievskyi, and Kyrylo Kuchynskyi. Blitzmask: Real-time instance segmentation approach for mobile devices. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 1799–1811. PMLR, 25–27 Apr 2023.

[3] Dan Ciresan, Alessandro Giusti, Luca Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[4] 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] 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. *CoRR*, abs/2010.11929, 2020.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[7] Ayan Gupta, Mayank Dixit, Vipul Kumar Mishra, Attulya Singh, and Atul Dayal. Brain tumor segmentation from mri images using deep learning techniques, 2023.

[8] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[9] Jonas Herzog. Adapt before comparison: A new perspective on cross-domain few-shot segmentation, 2024.

[10] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[11] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[12] Alexandre Kirchmeyer and Jia Deng. Convolutional networks with oriented 1d kernels, 2023.

[13] Rong Li, ShiJie Li, Xieyuanli Chen, Teli Ma, Juergen Gall, and Junwei Liang. Tfnet: Exploiting temporal cues for fast and accurate lidar semantic segmentation, 2024.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context, 2014.

[15] Mustafa Munir, William Avery, and Radu Marculescu. Mobilevig: Graph-based sparse attention for mobile vision applications, 2023.

[16] Ozan Oktay, Jo Schlemper, Loïc Le Folgoc, Matthew C. H. Lee, Mattias P. Heinrich, Kazunari Misawa, Kensaku Mori, Steven G. McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas. *CoRR*, abs/1804.03999, 2018.

[17] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012.

[18] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[19] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[21] 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. *CoRR*, abs/1609.05158, 2016.

[22] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net, 2023.

[23] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. *ECCV*, 2012.

[24] Shiyu Tang, Ting Sun, Juncai Peng, Guowei Chen, Yuying Hao, Manhui Lin, Zhihong Xiao, Jiangbin You, and Yi Liu. Pp-mobileseg: Explore the fast and accurate semantic segmentation model on mobile devices, 2023.

[25] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, abs/1908.00463, 2019.

[26] Sangmin Woo, Byeongjun Park, Hyojun Go, Jin-Young Kim, and Changick Kim. Harmonyview: Harmonizing consistency and diversity in one-image-to-3d, 2023.

[27] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. *CoRR*, abs/1807.10165, 2018.

[28] Jiafan Zhuang, Zilei Wang, Yixin Zhang, and Zhun Fan. Infer from what you have seen before: Temporally-dependent classifier for semi-supervised video segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3575–3584, June 2024.