OneNet: A Channel-Wise 1D Convolutional U-Net

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

Abstract

Many state-of-the-art computer vision architectures use U-Net for its adaptability and efficient feature extraction. However, the multi-resolution convolutional design often leads to significant computational demands, limiting deployment on edge devices. We present a streamlined alternative: a 1D convolutional encoder that retains U-Net's accuracy while enhancing its suitability for edge applications. Our novel encoder architecture achieves semantic segmentation through channel-wise 1D convolutions combined with pixel-unshuffle operations. By incorporating PixelShuffle, known for improving accuracy in super-resolution tasks while reducing computational load, OneNet captures spatial relationships without requiring 2D convolutions, reducing parameters by up to 47%. Additionally, we explore a fully 1D encoder-decoder that achieves a 71% reduction in size, albeit with some accuracy loss. We benchmark our approach against U-Net variants across diverse mask-generation tasks, demonstrating that it preserves accuracy effectively. Although focused on image segmentation, this architecture is adaptable to other convolutional applications. The code for the project is available at https://github.com/shbyun080/OneNet

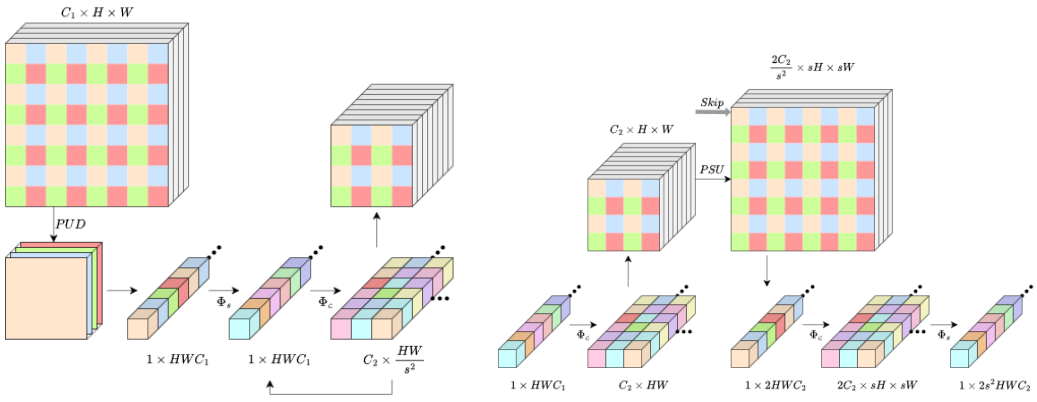
1 Introduction

With advancements in encoder-decoder architectures, the accuracy and versatility of vision models have reached unprecedented levels. However, deploying these high-parameter models on edge devices (e.g., mobile phones) poses a significant challenge due to their limited computational resources. Techniques like optimization and quantization have become essential to enable the use of state-of-the-art models on these devices.

Many modern architectures, including diffusion models [31, 35, 3], rely heavily on the U-Net [29] architecture as an encoder-decoder backbone. Yet, its structure is not optimized for efficiency in resource-constrained environments as they often overlook the inherent architectural inefficiencies. Since they typically employ a standard convolutional backbone such as ResNet [13], the parameter count can escalate rapidly, impacting efficiency and increasing the chance of overfitting. Although the architecture is highly adaptable, minimal research has been conducted to streamline its size for edge deployment. To solve this issue, we propose modifying the U-Net [29] backbone to reduce the number of parameters, thereby decreasing computing costs and model download size. This optimization would make U-Net [29] more feasible for edge deployment and open up possibilities for more complex models by reallocating resources to tasks of greater importance.

In contrast, areas like image super-resolution have long benefited from techniques like PixelShuffle [30], significantly improving pipeline efficiency without compromising spatial information. However, despite the clear advantages of these scaling techniques, they have not been widely explored in other domains. Additionally, while lightweight architectures like MobileNet [15] have been effective for more straightforward tasks such as classification, such structures remain underutilized in generative models. This knowledge gap suggests an opportunity to explore alternative efficiency-driven architectures for demanding vision tasks, potentially unlocking new performance levels and adaptability in model deployment.

Preprint. Under review.



(a) (b)

Figure 1: **Channel-Wise 1D Convolution Block** (a) Encoder convolution block with pixel-unshuffle downscaling replacing max pooling operation, followed by a single spatial and two channel-wise layers. (b) Decoder convolution block with pixel-shuffle upscaling for tensor upsampling, followed by a spatial layer between two channel-wise layers.

This paper proposes a novel adaptation of the U-Net [29] architecture that bridges the gap between state-of-the-art performance and edge-deployability by reducing the model size while preserving accuracy. Our approach is the first to leverage channel-wise 1D convolutions in conjunction with pixel-shuffling operations to enable efficient feature and spatial attention without reliance on 2D computations. By eliminating these operations that 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 lightweight deployment. Additionally, we optimize spatial processing by reducing kernel sizes, focusing instead on cross-feature interactions to further minimize memory requirements. This novel 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 while retaining spatial information through the introduction of pixel-unshuffle downscaling, pixel-shuffle upscaling, and channel-wise 1D convolutions.
- We implement two versions of U-Net [29] (1D encoder with 2D-decoder and 1D encoder-decoder) using our novel convolution block, effectively reducing the model size by 47% and 67% while maintaining reasonable accuracy.
- We evaluate our proposed method variations on multiple mask-prediction datasets and compare our results to commonly used backbones for U-Net to display performance retention while reducing the model's total size and the number of computations.

2 Related Works

U-Net [29, 36] introduces a new encoder-decoder approach to semantic segmentation, employing a technique of a contractive path followed by an expansive path. Developed for smaller mask-count datasets, it displays significant improvement over the existing state-of-the-art networks [6, 13]. One advantage of U-Net [29] is its ability to use a variety of backbones [13, 15, 7, 23] depending on the complexity of the task. U-Net [29] and its variations [36, 24] have become a cornerstone in the medical imaging field, especially for MRI image segmentation. Gupta et al. [12] utilizes U-Net [29] for brain tumor segmentation with notable success. Similarly, SAM-2 [28] extends this approach by introducing spatiotemporal mask predictions for videos. In contrast, Zhuang et al. [37] propose a novel idea to tackle the challenge of the lack of segmentation labels in video data by using a temporally dependent classifier (TDC) to mimic the human-like recognition procedure. However, despite these advances, standard U-Net [29] architectures remain computationally demanding for edge devices.

1D Convolutional Neural Networks [34, 19, 17] have been widely used for various special applications such as spectral MRI [34] and time-series [19] problems. As 1D calculations require fewer computations in most cases, they are preferred in most dimension-separable tasks. Additionally, recent research by Kirchmeyer et al. [17] demonstrates that a ConvNet consisting entirely of directional 1D convolutions performs comparably on ImageNet classification. Such results indicated that 1D convolutions can achieve similar performances on further complex tasks such as segmentation.

Pixel Shuffle [30] increases image resolution by converting low-resolution (LR) tensors into high-resolution (HR) images. The operation takes an $W \times H \times Cr^2$ tensor and converts it into a high-resolution $Wr \times Hr \times C$ tensor. In the field of image super-resolution, multiple state-of-the-art methods [5, 4] propose leveraging PixelShuffle [30] for efficient processing of images at a lower resolution without losing higher-resolution information. Our work is largely inspired by this technique, as it helps capture spatial relationships while reducing computational load.

Efficient segmentation models [32, 2, 22, 15, 20] have seen a rise in demand, leading to the development of models such as PP-MobileSeg [32], Blitzmask [2], or Mobilevig [22], which are designed specifically for mobile devices. Although PP-MobileSeg [32] 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 [20] 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.

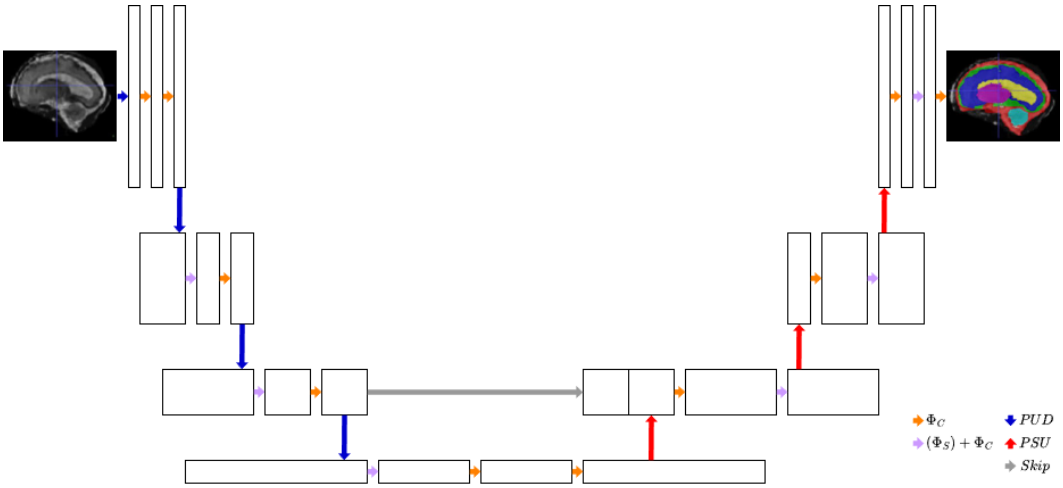


Figure 2: **Channel-Wise 1D Encoder-Decoder** OneNet employs a U-Net [29] architecture with skip connections for segmentation tasks. The architecture above is a 3-layer variant shown for simplicity. The encoder block replaces the max pool layer with pixel-unshuffle downscaling, with the image downscaled immediately on input for spatial relations to be captured. The decoder block replaces upsampling methods with a pixel-shuffle upscaling. In the architecture shown, 1D convolution is used for both encoder and decoder, with optional spatial convolution. To satisfy the spatial-preservation property in the decoder, we only decode to half resolution. The top layer of the decoder is implemented without batch normalization or ReLU to avoid zero-centering of the prediction head.

3 OneNet

We stated that our approach heavily relies on pixel-shuffling operations for the preservation of spatial information. This motivates us to treat the channel dimension as a spatial-augmented feature vector. We first cover the basis of pixel-unshuffle downscaling and its parameter-reduction property in section 3.1. Then, we explain our implementation of 1D convolutional blocks used to leverage the unique downsampling technique in section 3.1. Lastly, we elaborate on how U-Net [29] can be adapted to the proposed OneNet architecture through a breakdown of the encoder (section 3.3) and the decoder (section 3.4).

3.1 Pixel-Unshuffle Downscaling

We express the downsampling operation as a function $Y(m) = X(D(mS))$, where m is the output tensor index of Y, m is the total number of pixels in Y, S stands for the scaling factor, and D is the sampling strategy used, such as max pooling. The target of downsampling is to increase the receptive field of the network while decreasing the computation cost. Traditional convolutional neural networks achieve this through either max pooling or average pooling.

We propose to replace the pooling operation with pixel-unshuffle downscaling, which can be annotated as

$$D(X)_{i,a,b} = X_{\lfloor i/s^2 \rfloor, sa+\lfloor sa/i \rfloor, sb+sb(mod\ i)} \tag{1}$$

where the D denotes pixel-unshuffle downscaling operation, X is the input tensor, s is the scale of the downscaling, and $i \in [0, s^2C)$ where C is the number of channels in X. Although a scale greater than 2 is feasible, the resolution of the tensor is reduced by the factor of s^L where L is the number of layers. Thus, it would not leave sufficient room for downscaling for most datasets as $s = 3$ and $L = 4$ would downscale by the factor of 81 compared to $s = 2$ of 16.

For ease of discussion, we simplify the tensor with a single spatial dimension and a single channel dimension. We start with a tensor, with spatial information on column and channel on row dimension, as shown below

$$X = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \tag{2}$$

With the traditional 2D approach where the convolutional layer has a kernel size of $k = (2, 2)$ and stride of $s = 1$ to produce X', and the max pool layer D is applied with the assumption that the first element has the largest value, we get $X(D(mS)) = (a'_{00}, a'_{10})^T$. The number of multiplication calculations here would be the product of parameter size ($kC = 4$), number of convolutions ($MS/s = 2$), and number of output features $C = 2$, resulting in $kMSC^2 = 16$ operations.

In comparison, the proposed method would first perform pixel-unshuffle downscaling to attain $X' = (a'_{00}, a'_{01}, a'_{10}, a'_{11})^T$. Then, a 1D convolution is applied to directly reduce the number of channels to attain

$$X(D(mS)) = \begin{pmatrix} W_0 X' \\ W_1 X' \end{pmatrix} \tag{3}$$

which would result in the kernel size of $MC = 4$ and the number of output features $C = 2$, resulting in a total multiplication count of $MC^2 = 8$. This is half of the multiplication counts needed compared to the 2D approach.

3.2 1D-Kernel Convolution

Ignoring batch dimension for the sake of simplicity, traditional convolution block performs a 3D computation in spatial and channel dimensions, resulting in a parameter of (C_{in}, H, W, C_{out}), where C is the channel, H is the height, and W is the width of the input tensor. In depth-wise convolution proposed by Howard et al. [15], this weight is decomposed into spatial and channel dimensions, resulting in the parameter of $(H, W) + (C_{in}, C_{out})$.

Although MobileNet [15] implementation significantly reduces the number of parameters, it fails to capture spatial-channel dependencies due to its design. We show the convolutional steps taken by each method in fig. 3.

As pixel-unshuffle downscaling transfers spatial knowledge to the channel axis, we replace the 2D convolution layer with 1D convolution layers working on a flattened tensor of size (B, HWC). Channel-wise convolution processes this tensor and runs C_{out} 1D convolutions with the kernel size and stride of $k = s = C_{in}$ to attain a tensor of size (B, C_{out}, HW), effectively achieving parameter size of (C_{in}, C_{out}). As spatial information is still intact in the channel dimension, running a channel-wise convolution results in consideration of spatial-channel dependencies.

As spatial information solely depends on pixel-unshuffle downscaling, the scale factor also defines the model's receptive field, which is $Receptiveness = SL$. In the U-Net [29] architecture with OneNet encoder, the model would depend on the decoder for a larger receptive field.

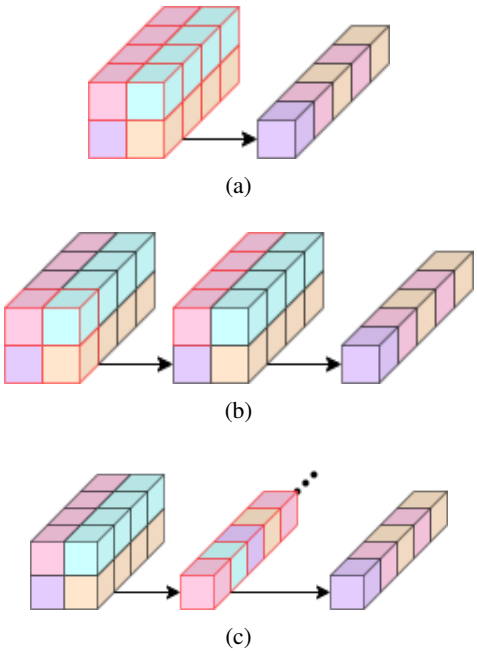


Figure 3: **Comparison of Convolutional Block** (a) Traditional 2D convolutional block with max pooling. (b) MobileNet [15] block with max pooling. (c) OneNet implementation with pixel-unshuffle downscaling followed by 1D convolution.

Algorithm 1 1D pixel-unshuffle downscaling

Input: X in shape $(B, C, H \times W)$, height H, width W **Output:** pixel-unshuffled X in shape $(B, H \times W \times C)$

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

3.3 1D Encoder

We show our proposed OneNet encoder block in fig. 1a, consisting of pixel-unshuffle downscaling, followed by two channel-wise 1D convolutions with an optional spatial convolution. One issue with pixel-unshuffling implementation is the required chain of flattening and shape-restoration steps. Thus, we propose a method for preserving all tensors in a single spatial dimension by implementing a 1D-compatible pixel-unshuffle algorithm. We show our adaptation in algorithm 1.

To compare the parameter size of the traditional 2D convolution block to the proposed OneNet encoder block, let us start with a tensor X of shape (C, H, W) after a downsampling operation for the 2D encoder. With the 2D encoder, following fig. 3c, the two convolutions would have weight of shapes $(C, k, k, 2C)$ and $(2C, k, k, 2C)$. With a stride of 1 and a kernel size $k = 3$, this results in multiplication counts of $2k^2C^2HW$ and $4k^2C^2HW$, totaling $6k^2C^2HW$ calculations. For OneNet, the input tensor would have the shape of $(4C, H, W)$ due to pixel-unshuffle downscaling, giving weights of shapes $(4C, 2C)$ and $(2C, 2C)$. This gives the total multiplication count of $8C^2HW + 4C^2HW = 12C^2HW$. Thus, the ratio of parameters for a single block would be $(6k^2C^2)/(12C^2) = k^2/2$. As k must be at least 2, this would result in a smaller computation and parameter size.

Method	VOC [10]			PET_F [25]			PET_S [25]			MSD Heart [1]			MSD Brain [1]			MSD Lung [1]		
	$\mathcal{L}_{CE}$	$mAP_{0.5}$	mIOU	$\mathcal{L}_{CE}$	$mAP_{0.5}$	mIOU	$\mathcal{L}_{CE}$	$mAP_{0.5}$	mIOU	$\mathcal{L}_{CE}$	$mAP_{0.5}$	mIOU	$\mathcal{L}_{CE}$	$mAP_{0.5}$	mIOU	$\mathcal{L}_{CE}$	$mAP_{0.5}$	mIOU
U-Net_4 [29]	1.985	0.0541	0.182	2.206	0.0329	0.316	0.243	0.7717	0.713	0.0223	0.4305	0.063	0.0455	0.2450	0.001	0.0010	0.5542	**0.009**
$ResNet_{34}$ [13]	1.321	0.0806	0.332	**0.648**	0.0503	0.597	0.179	0.9098	0.801	0.0087	0.4375	0.065	0.1305	0.4188	0.079	0.0010	0.5529	0.009
$ResNet_{50}$ [13]	**1.079**	**0.0921**	**0.372**	1.027	**0.0512**	**0.599**	**0.189**	**0.9303**	**0.815**	0.0086	0.4368	0.065	0.0496	0.5419	0.010	**0.0007**	0.5566	0.009
MobileNet [15]	2.007	0.0480	0.166	2.386	0.0329	0.252	0.262	0.6091	0.664	0.0288	0.3265	0.047	**0.0351**	0.5764	0.011	0.0010	**0.5787**	0.008
$OneNet_{e,4}$	2.144	0.0485	0.160	2.713	0.0279	0.216	0.309	0.5176	0.636	**0.0041**	**0.4396**	**0.066**	0.0363	**0.5789**	**0.105**	0.0007	0.5531	0.009
$OneNet_{ed,4}$	2.553	0.0363	0.149	3.080	0.0227	0.172	0.437	0.2179	0.535	0.0076	0.4187	0.062	0.0455	0.5415	0.099	0.0009	0.5510	0.008

Table 1: **Baseline Comparisons on Semantic Segmentation Datasets** U-Net and OneNet are trained on datasets [10, 25, 1] without pre-training for a fair comparison, as outlined by the original U-Net [29]. ResNet [13] and MobileNet [15] are pre-trained on ImageNet-1K for accuracy comparison. PET_F and PET_S stand for the full and small mask versions of the Oxford Pet [25] dataset with 38 and 3 classes each. U-Net_i stands for vanilla U-Net [29] encoder with downsampling layers of i. $Resnet_i$ stands for i-layer version of ResNet [13]. $OneNet_{e(d),i}$ has i downsampling layers, with e and d each standing for encoder and decoder replaced with the proposed architecture, respectively. The best results are in **bold**.

Method	# Param (M)	Param (MB)	FLOPS (GB)	Memory (MB)
U-Net_4 [29]	31.04	124.03	104.72	509.61
U-Net_5 [29]	124.42	497.41	130.80	524.29
$ResNet_{34}$ [13]	25.05	98.07	29.40	**241.17**
$ResNet_{50}$ [13]	74.07	287.83	84.98	450.36
MobileNet [15]	14.40	57.47	83.96	671.09
$OneNet_{e,4}$	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**	**22.92**	799.01
$OneNet_{ed,5}$	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. With the 4-layer model, OneNet achieves 47% reduction with encoder swap and 69% reduction with encoder-decoder swap. 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 [29] encoder with downscaling layers of i. $Resnet_i$ stands for i-layer version of ResNet [13]. $OneNet_{e(d),i}$ has i downscaling layers, with e and d each standing for encoder and decoder, respectively.

Additionally, we test an optional spatial convolution that works on a flattened 1D tensor of size (CHW) with a kernel size $k << C$ and stride $s = 1$. We do not test our model on a pre-trained encoder as per U-Net [29] implementation. As the paper focuses on the backbone potential of 1D convolutions, pre-training is out of the scope of our paper.

3.4 Segmentation Decoder

Additionally, we design and propose a decoder block (fig. 1b) that performs upsampling blocks through pixel-shuffle upscaling. Unlike the OneNet encoder, we perform an upscaling operation, which moves information from channel to spatial instead. Thus, the spatial information is not much represented in the channel dimension. Despite this difference, if the OneNet decoder is used in conjunction with the proposed encoder, we can make a mild assumption that the channel dimension already consists of sufficient spatial information due to the unshuffling steps performed. However, this signifies that although the encoder can be used in conjunction with any decoder architecture, the OneNet decoder is strictly specific to the OneNet encoder.

Otherwise, the decoder follows a standard U-Net [29] architecture with skip connections with a similar block as the proposed decoder. As the upscaling method reduces the channel size to one-fourth, there is no need to change the channel count after the concatenations from skip connections.

Method	PET$_S$ [25]				MSD Heart [1]				MSD Lung [1]			
	$\mathcal{L}_{CE}$	mAP$_{0.5}$	mIOU	DSC	$\mathcal{L}_{CE}$	mAP$_{0.5}$	mIOU	DSC	$\mathcal{L}_{CE}$	mAP$_{0.5}$	mIOU	DSC
OneNet$_{e,4}$-S	0.320	**0.5275**	0.632	0.963	**0.0034**	0.4383	0.065	**0.699**	**0.0006**	0.5510	**0.008**	**0.115**
OneNet$_{e,4}$-C	**0.309**	0.5176	**0.636**	**0.967**	0.0363	**0.4386**	**0.066**	0.605	0.0007	**0.5531**	0.009	**0.115**
OneNet$_{ed,4}$-S	0.416	0.2253	0.529	0.868	0.0125	0.3938	0.059	0.544	0.0007	0.5512	**0.008**	0.112
OneNet$_{ed,4}$-C	0.437	0.2179	0.535	0.842	0.0076	0.4187	0.062	0.605	0.0007	0.5510	**0.008**	0.111

Table 3: **Ablation Study of Spatial Convolution** We study the effect of spatial convolution before channel convolution for datasets with small segmentation mask counts [25, 1]. The best results are in **bold**. We see an overall performance to be unaffected by the addition of spatial convolution with a kernel size of 9 for both encoder-only and encoder-decoder replacements.

4 Experiments

We implement the OneNet architecture using PyTorch [26] and optimize it with the Adam [16] optimizer, setting an initial learning rate of 1×10^{-4}. We set a learning rate scheduler decaying the learning by a factor of 0.1 every 20 epochs starting from epoch 50. The model is trained for a total of 300 epochs, with batch sizes of 32 or 64 depending on memory availability, using an input image resolution of 512×512 and a segmentation mask resolution of 256×256. We omit batch normalization and ReLU activation from the top decoder layer to ensure accurate segmentation predictions, avoiding any potential zero-centering effects on the output. The initial bottleneck channel count is set to 64, while the kernel size for the optional spatial convolution layer is set to 9. The pixel-unshuffling downscaling is applied with a scale factor of 2, resulting in $16\times$ total downscaling factor for a 4-layer encoder. Training is conducted on a single RTX 4090 GPU with 24GB of VRAM.

4.1 Mask Prediction Tasks

We evaluate our method across six diverse datasets: PASCAL VOC [10], Oxford Pet with breed masks and subject masks [25], and three medical segmentation datasets—MSD Heart, MSD Brain, and MSD Lung [1]. These datasets represent a broad spectrum of masking complexities, with class counts ranging from 38 in Oxford Pet with breed masks to as few as 2 for MSD Lung, and intermediate counts of 21, 3, 4, and 4 for the other datasets. Consistent with Ronneberger et al. [29], we do not pre-train the encoder for both U-Net [29] and OneNet models to provide a controlled comparison, allowing us to assess OneNet's performance without the advantages of transfer learning. However, for models such as ResNet [13] and MobileNet [15], which typically benefit from large-scale pre-training, we initialize them on ImageNet-1K [8] and then fine-tune them on each specific dataset. This provides a nuanced view of OneNet's capabilities in comparison to commonly used models in the field. Our accuracy evaluation, detailed in table 1, uses U-Net [29] as a primary baseline to ensure a fair and relevant comparison across architectures.

Training setup and evaluation metric As all datasets explored are multi-class segmentation tasks, cross-entropy loss is used with the weight of the background mask reduced to a quarter. We report cross-entropy loss ($\mathcal{L}_{CE}$), mean average precision with an IOU threshold of 0.5 (mAP$_{0.5}$), and mean intersection-over-union (mIOU).

Analysis on medical tumor segmentation For all medical tumor detection datasets [1], table 1 indicates that the proposed OneNet model performs on par with established baseline architectures. Notably, for the MSD Heart dataset [1], OneNet shows a slight edge in accuracy, achieving a 2% improvement over the baseline models. This suggests that OneNet can maintain competitive performance even in specialized applications, capturing important features required for medical imaging.

Additionally, OneNet achieves model size reduction while maintaining an accuracy drop within 1% across datasets, accompanied by a substantial 47% reduction in model parameters. This underscores OneNet's ability to perform complex segmentation tasks with a far more compact architecture. Such efficiency makes it particularly well-suited for applications involving smaller-mask prediction tasks, including medical segmentation and tasks requiring precise depth or normal map generation.

The ability to sustain high accuracy with fewer parameters means that OneNet is optimized for both performance and resource constraints, making it an excellent candidate for real-time or edge-based medical imaging applications. By minimizing computational demands without a significant compromise in accuracy, OneNet offers a practical, scalable solution for high-precision tasks in medical diagnostics and other specialized segmentation applications.

Analysis on general multi-class segmentation The results in table 1 show an 11% and 15% drop in accuracy for the PASCAL VOC [10] and full-size Oxford Pet [25] datasets, respectively, with a 10% decrease for the Oxford Pet dataset with fewer masks for encoder-only replacement. These declines highlight a trade-off between accuracy and model size, especially when working with many classes. However, we consider this trade-off acceptable to enable edge deployability where resources are limited.

This pattern suggests that the receptive field created in OneNet encoder-decoder by pixel-unshuffle downscaling (with a scale factor of 2) effectively enhances local feature detection rather than image-wide classification. Thus, encoder-only replacement would be better suited for such tasks requiring a large receptive field, supporting the two proposed models' efficiency without overly compromising performance, making it suitable for resource-constrained deployments.

4.2 Impact on Model Size

In table 2, we present the parameter size and FLOPs of OneNet compared to baseline models [29, 13, 15], using a sample input tensor of size $(1, 3, 256, 256)$ for baselines and $(1, 3, 512, 512)$ for OneNet. OneNet achieves substantial efficiency gains, with approximately 47%/25% reductions in parameters and FLOPs for its encoder and 67%/78% reductions for the complete encoder-decoder structure. These outcomes align closely with the theoretical calculations in section 3.1 and demonstrate OneNet's capability to handle high-resolution inputs while significantly reducing computational demands.

The compact size of OneNet is particularly noteworthy, as it can reach below 40MB—a 71% reduction compared to the standard U-Net. This drastic reduction makes OneNet highly suitable for deployment on edge devices, allowing the model weights to be downloaded and stored locally with minimal storage overhead. However, we observe an increase in memory usage due to the pixel-unshuffle downscaling, which increases the channel count by a factor of 4 compared to the constant channel dimension in max pooling. While this higher channel count imposes some memory cost, it enhances the model's ability to capture fine-grained spatial information, ultimately improving feature representation without compromising overall efficiency.

4.3 Ablation on Spatial Convolutions

Further study of the effect of channel convolutions in the OneNet convolution blocks is shown in table 3. To discuss whether spatial convolution similar to that of a MobileNet [15] would influence the model, we add the spatial convolutions to the 4-layer encoder and encoder-decoder OneNet replacements as shown in fig. 2, with kernel size 9 and stride 1. We additionally report the Dice coefficient (DSC) for pixel-wise agreement. The results suggest that due to the pixel-unshuffle downscaling's organization of the pixels in a convolutional manner without repetition, such grouping drives the spatial layer to have minimal effect. This supports channel-wise convolution's sufficiency in capturing spatial information as the addition of a spatial module, albeit small, does not affect the model's accuracy in any way.

Further analysis of the role of channel-wise convolutions within OneNet's convolution blocks is presented in table 3. To examine whether spatial convolutions, similar to those in MobileNet [15], would impact model performance, we incorporate spatial convolutions into the OneNet 4-layer encoder and encoder-decoder structures, as depicted in fig. 2, using a kernel size of 9 and stride of 1. Alongside this configuration, we report the Dice coefficient (DSC) to measure pixel-wise agreement and assess segmentation accuracy.

With an accuracy difference in the range of 2%, the results indicate that the pixel-unshuffle downscaling technique—which organizes pixels in a structured convolutional manner without redundancy—minimizes the need for spatial convolutions. This design allows the channel-wise convolution to effectively capture spatial information independently. The addition of the spatial convolution module had a negligible impact on accuracy, underscoring that OneNet's channel convolutions are

sufficient for spatial feature extraction. Thus, these findings validate the efficiency of OneNet's architecture by demonstrating that a lightweight, channel-focused approach is capable of high spatial accuracy without the added complexity of a spatial convolution layer.

5 Future Work

The paper introduces a novel architecture, OneNet, which efficiently combines channel-wise 1D convolutions with pixel-unshuffle operations for semantic segmentation. **CLIP** [27], known for its ability to learn visual representations from natural language supervision, offers a promising direction for integration with OneNet. Incorporating CLIP with OneNet can enable semantic segmentation using multimodal embeddings. For instance, **CLIP-ES** [21] demonstrates the potential of using CLIP as a segmenter and leveraging weakly-supervised learning to generate segmentation masks. Extending OneNet to leverage such advancements could broaden its applicability across tasks and domains. Additional approaches for multimodal learning and segmentation can be found in [33, 11, 18].

Furthermore, exploring pretraining and fine-tuning strategies can significantly enhance the adaptability and performance of OneNet. Recent works on transformer architectures, such as Dosovitskiy et al. [9], highlight the effectiveness of pretraining on large-scale datasets followed by fine-tuning on mid-sized or small benchmarks. Similarly, Woo et al. [14] showcase the benefits of cross-domain adaptation, focusing on robust feature representations with limited data. In alignment with these strategies, pretraining OneNet on a general dataset and fine-tuning it for specific tasks (e.g., medical imaging or autonomous systems) would optimize its performance across varied applications, especially in resource-constrained settings.

6 Conclusion

We presented OneNet, a novel architecture that effectively reduces the number of parameters required while preserving accuracy through the use of 1D convolutions. Our approach excels in local-feature-centric mask-generating tasks by relying solely on channel-wise 1D convolutions, ensuring that spatial attention is maintained through the use of pixel-unshuffle downscaling. In contrast to existing models that require large parameter counts due to an increasing number of channels and constant kernel sizes, our method leverages channel-wise features to significantly reduce model size without compromising performance. We demonstrate that pixel-unshuffle downscaling is both efficient and information-preserving, allowing for stable feature transfer between scales without the need for traditional 2D convolutions. Additionally, we show that the concept of pixel shuffling can be effectively applied to decoder networks by assuming spatial dependencies. Looking forward, we are keen to explore various pixel-repositioning techniques across different architectures, aiming to further advance the development of efficient AI models that can deliver high performance with reduced computational overhead.

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. *arXiv:2106.05735*, 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] Sanghyun Byun, Jacob Song, and Woo Seong Chung. Multidepth: Multi-sample priors for refining monocular metric depth estimations in indoor scenes. *arXiv:2411.01048*, 2024.

[4] Soumick Chatterjee, Alessandro Sciarra, Max Dünnwald, Raghava Vinaykanth Mushunuri, Ranadheer Podishetti, Rajatha Nagaraja Rao, Geetha Doddapaneni Gopinath, Steffen Oeltze-Jafra, Oliver Speck, and A. Nürnberger. Shuffleunet: Super resolution of diffusion-weighted mris using deep learning. *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 940–944, 2021.

[5] Zheng Chen, Zongwei Wu, Eduard Zamfir, Kai Zhang, Yulun Zhang, Radu Timofte, Xiaokang Yang, Hongyuan Yu, Cheng Wan, Yuxin Hong, Zhijuan Huang, Yajun Zou, Yuan Huang, Jiamin Lin, Bingnan Han, Xianyu Guan, Yongsheng Yu, Daoan Zhang, Xuanwu Yin, Kunlong Zuo, Jinhua Hao, Kai Zhao, Kun Yuan, Ming Sun, Chao Zhou, Hongyu An, Xinfeng Zhang, Zhiyuan Song, Ziyue Dong, Qing Zhao, Xiaogang Xu, Pengxu Wei, Zhi-chao Dou, Gui-ling Wang, Chih-Chung Hsu, Chia-Ming Lee, Yi-Shiuan Chou, Cansu Korkmaz, A. Murat Tekalp, Yubin Wei, Xiaole Yan, Binren Li, Haonan Chen, Siqi Zhang, Sihan Chen, Amogh Joshi, Nikhil Akalwadi, Sampada Malagi, Palani Yashaswini, Chaitra Desai, Ramesh Ashok Tabib, Ujwala Patil, Uma Mudenagudi, Anjali Sarvaiya, Pooja Choksy, Jagrit Joshi, Shubh Kawa, Kishor Upla, Sushrut Patwardhan, Raghavendra Ramachandra, Sadat Hossain, Geongi Park, S. M. Nadim Uddin, Hao Xu, Yanhui Guo, Aman Urumbekov, Xingzhuo Yan, Wei Hao, Minghan Fu, Isaac Orais, Samuel Smith, Ying Liu, Wangwang Jia, Qisheng Xu, Kele Xu, Weijun Yuan, Zhan Li, Wenqin Kuang, Ruijin Guan, Ruting Deng, Zhao Zhang, Bo Wang, Suiyi Zhao, Yan Luo, Yanyan Wei, Asif Hussain Khan, Christian Micheloni, and Niki Martinel. NTIRE 2024 Challenge on Image Super-Resolution (×4): Methods and Results . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 6108–6132, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.

[6] 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.

[7] Soham De and Samuel L. Smith. Batch normalization biases residual blocks towards the identity function in deep networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[8] 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.

[9] 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.

[10] 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.

[11] Shi-Cheng Guo, Shang-Kun Liu, Jing-Yu Wang, Wei-Min Zheng, and Cheng-Yu Jiang. Clip-driven prototype network for few-shot semantic segmentation. *Entropy*, 25:1353, 09 2023.

[12] Ayan Gupta, Mayank Dixit, Vipul Kumar Mishra, Attulya Singh, and Atul Dayal. Brain tumor segmentation from mri images using deep learning techniques. *arXiv:2305.00257*, 2023.

[13] 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.

[14] Jonas Herzog. Adapt before comparison: A new perspective on cross-domain few-shot segmentation. *arXiv:2402.17614*, 2024.

[15] 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.

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

[17] Alexandre Kirchmeyer and Jia Deng. Convolutional networks with oriented 1d kernels. *arXiv:2309.15812*, 2023.

[18] Taha Koleilat, Hojat Asgariandehkordi, Hassan Rivaz, and Yiming Xiao. Medclip-sam: Bridging text and image towards universal medical image segmentation. *arXiv:2403.20253*, 2024.

[19] Dongyang Kuang. A 1d convolutional network for leaf and time series classification. *arXiv:1907.00069*, 2020.

[20] Rong Li, ShiJie Li, Xieyuanli Chen, Teli Ma, Juergen Gall, and Junwei Liang. Tfnet: Exploiting temporal cues for fast and accurate lidar semantic segmentation. *arXiv:2309.07849*, 2024.

[21] Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. *arXiv:2212.09506*, 2023.

[22] Mustafa Munir, William Avery, and Radu Marculescu. Mobilevig: Graph-based sparse attention for mobile vision applications. *arXiv:2307.00395*, 2023.

[23] Khanh-Binh Nguyen, Jaehyuk Choi, and Joon-Sung Yang. Eunnet: Efficient un-normalized convolution layer for stable training of deep residual networks without batch normalization layer. *IEEE Access*, 2023.

[24] 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.

[25] 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.

[26] 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.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[28] 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. *arXiv:2408.00714*, 2024.

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

[30] 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.

[31] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. *arXiv:2309.11497*, 2023.

[32] 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. *arXiv:2304.05152*, 2023.

[33] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. *arXiv:2111.15174*, 2022.

[34] Zi Wang, Chen Qian, Di Guo, Hongwei Sun, Rushuai Li, Bo Zhao, and Xiaobo Qu. One-dimensional deep low-rank and sparse network for accelerated mri. *IEEE Transactions on Medical Imaging*, 42(1):79–90, 2023.

[35] Sangmin Woo, Byeongjun Park, Hyojun Go, Jin-Young Kim, and Changick Kim. Harmonyview: Harmonizing consistency and diversity in one-image-to-3d. *arXiv:2312.15980*, 2023.

[36] 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.

[37] 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.