A lightweight Depthwise Separable Dilated Multires Network (DDMnet) for Instance Segmentation

Nikolaos Detsikas[1], Christos Chatzisavvas[1], Nikolaos Mitianoudis[1*], Ioannis Pratikakis[1]

[1*]Department of Electrical and Computer Engineering, Democritus University of Thrace, University Campus Xanthi-Kimmeria, Xanthi, 67100, Greece.

*Corresponding author(s). E-mail(s): nmitiano@ee.duth.gr;
Contributing authors: ndetsika@ee.duth.gr; cchatzis@ee.duth.gr; ipratika@ee.duth.gr;

Abstract

The differentiation and identification of individual objects within a scene is a fundamental problem in modern computer vision. This task, known as instance segmentation, has received extensive attention in recent years due to the advances in Deep Learning. Despite notable progress, many state-of-the-art approaches rely on increasingly complex architectures with high computational requirements, making them unsuitable for real-time applications, such as autonomous driving. In this work, we propose a compact model for addressing the instance segmentation problem, that integrates a novel convolutional encoder, named DDMnet, with the Mask2Former segmentation framework. The proposed network achieves low computational complexity and delivers performance close to state-of-the-art methods on benchmark datasets, including Cityscapes and ADE20K, demonstrating its potential for real-time deployment.

Keywords: Cityscapes, ADE20K, Instance Segmentation, Deep Learning

1 Introduction

Instance segmentation, a task that combines object detection with semantic segmentation, has emerged as a critical component in various computer vision applications, such as autonomous vehicles, medical imaging, and augmented reality. This task involves identifying and annotating each object instance in an image separately and presents significant challenges due to its requirement for precise pixel-level predictions and accurate object bounding box identification.

The duality of the instance segmentation problem puts much larger demands on the network size, in order for an architecture to achieve acceptable performance. Traditional approaches for instance segmentation often rely on large, computationally intensive models, with heavy backbone networks and highly complex feature extractors, designed to maximize accuracy on benchmark datasets, such as ADE20K or Cityscapes. However, these models are often unsuitable for deployment on edge devices with hardware constraints, such as smartphones, drones, IoT devices or even self-driving cars, where the installation of the necessary hardware and power supply is prohibiting.

The need for real-time processing on hardware-limited devices has led to a growing interest in the development of lightweight instance segmentation models. Such models must achieve a strict balance between computational efficiency and prediction accuracy. Existing lightweight architectures for related tasks, such as object detection, demonstrate the feasibility of optimizing for speed and resource usage. However, incorporating instance-level segmentation into these models remains challenging due to the increased complexity and higher dimensionality of mask predictions.

In this paper, the authors address the gap by proposing a small-scale, real-time instance segmentation model tailored for devices with hardware limitations. Our work builds on recent achievements in efficient neural architectures in the field of image segmentation. Strategies are investigated in order to reduce computational burden, while maintaining high segmentation accuracy. The main focus is on compact convolutional feature extractor structures that can be integrated into the Mask2Former framework, that result in a highly practical and competitive tradeoff between performance and network size

The proposed model is evaluated on standard instance segmentation benchmarks that reflect real-world constraints faced by edge devices. Our approach demonstrates competitive performance in terms of both accuracy and speed, achieving real-time inference on hardware-constrained platforms, such as Raspberry Pi and mobile GPUs. This work highlights the potential for deploying instance segmentation models in scenarios previously deemed impractical due to hardware limitations, paving the way for broader adoption of computer vision solutions in resource-limited environments.

To get an insight on current instance segmentation developments, the current state-of-the-art models are reviewed with the main focus on those with small computational complexity. Zhang et al. [1] introduced OpenSeeD, a unified framework for open-vocabulary object detection and segmentation, leveraging vision-language models to align image features with textual descriptions of classes. Their approach eliminates the need for extensive task-specific annotations, enabling the model to generalize effectively to unseen classes. By employing pre-trained vision-language backbones, such

as CLIP, the framework simplifies the architecture while maintaining competitive performance on open-vocabulary benchmarks. This work highlights the potential of vision-language alignment for scalable and efficient object detection and segmentation in diverse scenarios.

Jain et al. [2] introduced OneFormer, a transformer-based framework designed for universal image segmentation tasks, encompassing semantic, instance, and panoptic segmentation. The model unifies these tasks within a single architecture, leveraging task-conditioned prompts to adapt its predictions dynamically. By utilizing a shared backbone and transformer decoder, OneFormer achieves state-of-the-art performance across multiple segmentation benchmarks, demonstrating its versatility and efficiency in handling diverse segmentation challenges.

Rossi et al. [3] proposed Self-Balanced R-CNN, an enhanced instance segmentation framework addressing class imbalance issues prevalent in object detection and segmentation tasks. The approach introduces a self-balancing mechanism that adaptively reweights class contributions during training, improving the representation of minority classes without adding significant computational overhead. By integrating this mechanism into the R-CNN pipeline, the method achieves superior performance on benchmarks with imbalanced datasets, demonstrating its efficacy in producing accurate segmentations across diverse classes.

Zhang et al. [4] addressed the limitations of Vision Transformers (ViTs) in handling small datasets by introducing a lightweight Depth-Wise Convolution module that complements the ViT architecture. While ViTs excel at capturing global information, they often overlook fine-grained local details, which convolutional neural networks (CNNs) inherently capture through local filters. The proposed Depth-Wise Convolution module serves as a shortcut to bypass certain Transformer blocks, enabling the model to efficiently capture both local and global information with minimal computational overhead.

Kirillov et al. [5] introduced the Segment Anything Model (SAM), a vision model designed for universal segmentation tasks. SAM leverages a combination of an image encoder, prompt encoder, and mask decoder, achieving significant advancements in segmentation by making models more flexible and capable of segmenting diverse object categories. Their work builds upon previous efforts in generalizable segmentation models, offering a more scalable solution for a wide range of segmentation tasks with minimal task-specific tuning, enabling real-time and versatile segmentation in varied environments.

Zhang et al. [1] presented the Mask-Piloted Transformer, a model that is built on Mask2Former, with a variety of backbones, and feeds noisy ground-truth masks into decoder layers next to the learnable queries for reducing segmentation inconsistency. This "mask-piloted" strategy is applied only at training time without adding significant overhead during inference.

Finally, Cheng et al. [6] proposed the Masked-attention Mask Transformer (M2T), a novel architecture for universal image segmentation tasks. The M2T model integrates a masked-attention mechanism into the Transformer framework, enabling it to effectively handle both global context and fine-grained details necessary for segmentation. By leveraging this attention mechanism, M2T achieves superior performance

across a range of segmentation benchmarks, including instance, semantic, and panoptic segmentation. This work builds upon previous transformer-based segmentation model backbones, that act as feature extractors, enhancing their flexibility and generalization capabilities.

In this paper, the Mask2Former architecture is adapted with the inclusion of the proposed DDMnet encoder-backbone that reduces the computational complexity without reducing its performance. The DDMnet features multi-resolutional blocks with depthwise and dilated convolutional blocks that keep the computational load low without losing much of the previous performance.

2 The Proposed DDMnet

In this section, an in-depth analysis of the design details and architectural specifications of our proposed model is provided, the Depthwise Separable Dilated Multires Network (*DDMnet*). In addition, the key components that constitute DDMnet are discussed, highlighting how its design leverages depthwise separable convolutions, dilated convolutions, and MultiRes blocks to achieve efficient and accurate feature extraction.

The implementation code of DDMnet in PyTorch is available online [1].

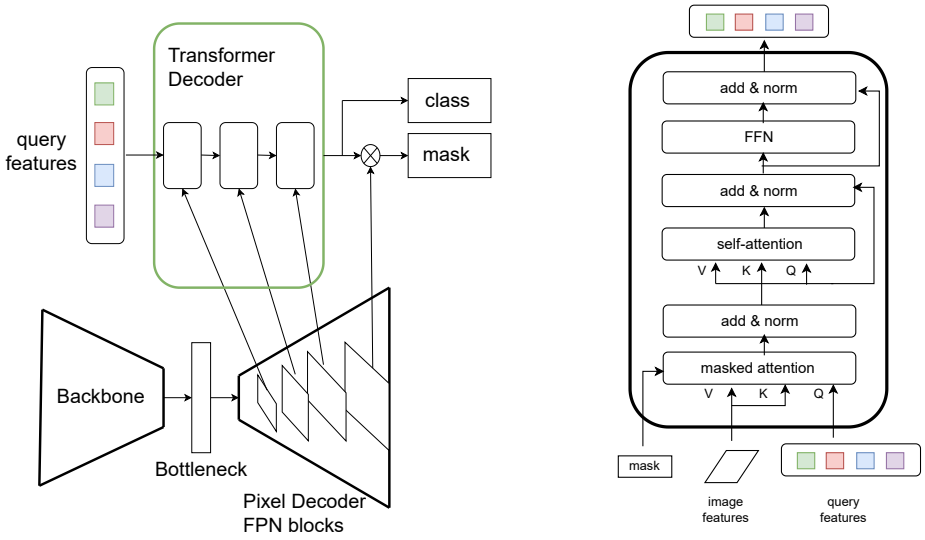
2.1 Mask2Former - The baseline architecture

The Mask2Former [6] is a transformer-based architecture designed for universal image segmentation tasks, which includes instance segmentation, semantic segmentation, and panoptic segmentation. It extends the ideas introduced in MaskFormer [7] by incorporating a more refined approach to learning and predicting segmentation masks. Figure 1 illustrates the Mask2Former architecture.

This framework model consists of several architectural blocks that perform the image segmentation tasks. Here is an overview of the basic blocks:

- *Backbone*: A typical convolutional/transformer backbone (i.e. ResNet, Swin Transformer) extracts feature maps from the input image.
- *Pixel decoder*: The pixel decoder generates multi-scale features by processing the backbone outputs. In a manner similar to that of a standard U-Net [8] flow, the pixel encoder produces high resolution features, starting from the low resolution backbone outputs and moving on to the higher resolution outputs. In addition to that, it provides multiscale output representations, similar to those of a Feature Pyramid Network (FPN) [9], which are later used by the transformer decoder.
- *Transformer decoder with masked attention*: The transformer decoder layer consists of multiple transformer decoder blocks, each processing one of the pixel decoder outputs. In contrast to a standard transformer decoder block, the masked-attention flavour of Mask2Former first uses cross-attention to attend to the foreground region of the input query only, instead of the whole feature map.

[1] https://github.com/detsikas/DDMnet



(a) The Mask2Former architecture [6]

(b) A transformer decoder block with masked attention

Fig. 1: Overview of the Mask2Former architecture and its transformer decoder block.

2.2 DDM backbone encoder

In this paper, the Mask2Former architecture basis is used with our introduced DDM encoder as the backbone component. The encoder is inspired from our previous endeavour, the DMVA network architecture [10], where we used dilated convolutions, multires and visual attention blocks in order to achieve state-of-the-art performance in the Document Binarization task. Figure 2 shows the original DMVAnet architecture used for the Document Binarization task.

This study uses some of these encoder features and applied modifications to improve performance while maintaining low computational complexity. Thus, in this work, the developed DMVAnet encoder is adapted, in order to suit the needs of instance segmentation.

The main architectural traits that are used in the proposed DDM encoder are briefly described:

– *Dilated convolutions*: Introduced in the DeepLab network series [11–13], they combat the negative effects of consecutive pooling layers (spatial resolution loss), such as those in a U-Net convolutional encoder. Dilated (a-trous) convolutions enlarge the receptive field of filters and provide dense extracted features without excessive memory requirements.

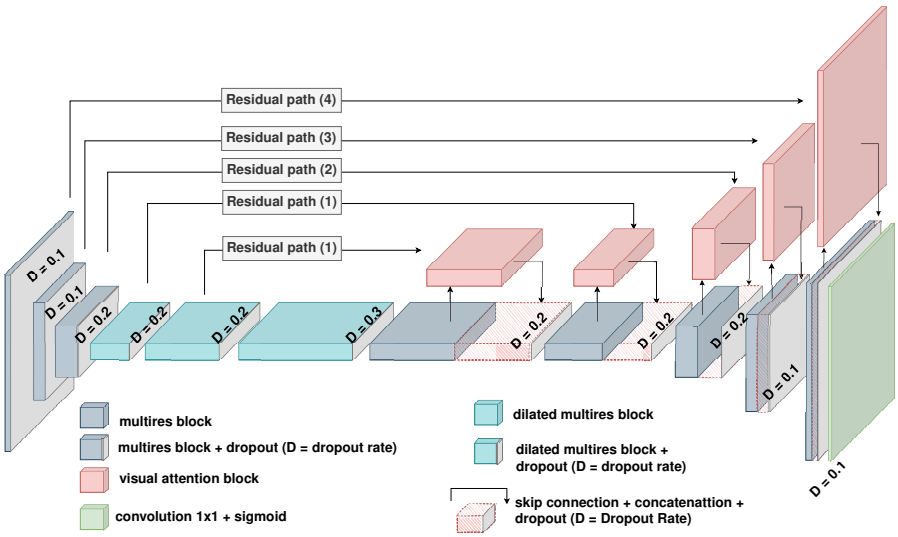


Fig. 2: DMVAnet overall architecture. The Residual path, the Visual attention block and the MultiRes block architecture are shown in Fig. 3, 4 and 5 respectively.

– *MultiRes block*: Presented by Ibtehaz and Rahman in [14], the MultiRes block replaces the Inception blocks of higher kernel resolution, such as 5×5 and 7×7 with a series of 3×3 kernels with residual connections. Consequently, memory requirements and computational complexity are kept low, without dropping prediction accuracy values. Figure 5 illustrates the MultiRes block architecture. The MultiRes block is further analyzed in the analysis of the model encoder later in this section.

All basic convolutional blocks have also been replaced with *depthwise separable convolutions*. Depthwise separable convolutions (see figure 6) are an efficient variant of standard convolution operations commonly used in deep learning, particularly in lightweight models, such as MobileNet [15] and Effiecient Net[16]. They have also been used in other popular networks, such as Xception [17], Mobilenet v2 [18]. They decompose the standard convolution into two separate operations:

– *Depthwise Convolution*: A convolution is performed on each input channel independently with a single convolutional filter kernel. Spatial information is captured here per channel.
– *Pointwise Convolution*: A 1×1 convolution is applied to combine the output of the depthwise convolutions across channels. Spatial information is mixed to construct the final feature map.

By using depthwise separable convolutions, modern architectures achieve a good balance between accuracy and computational efficiency, making them suitable for deployment in real-time and resource-constrained scenarios.

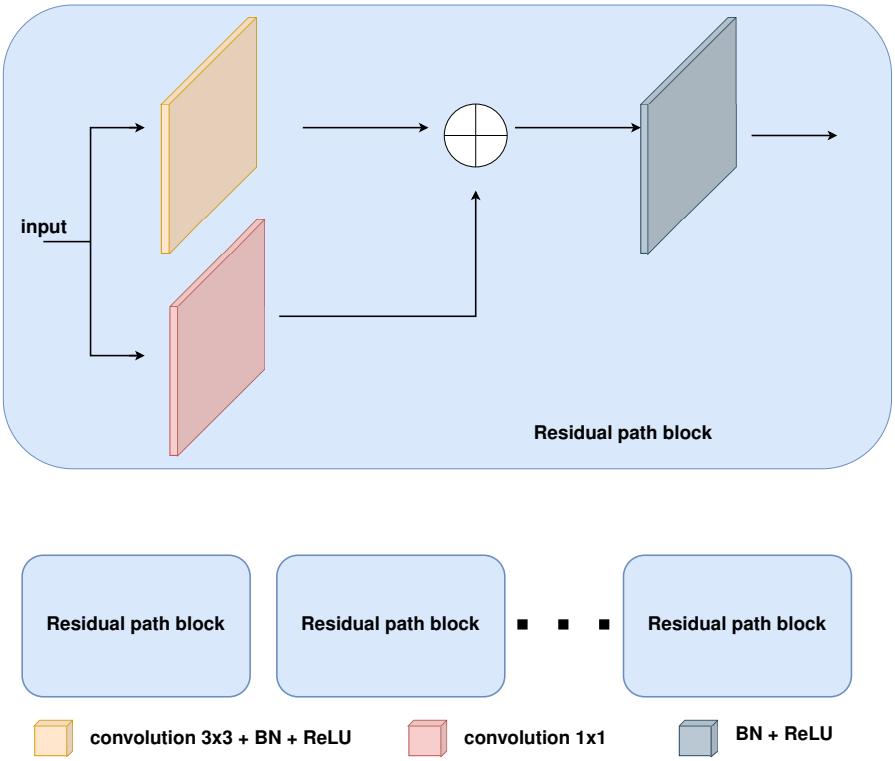


Fig. 3: The Residual path block is depicted in more detail.

Here is the mathematical formulation of the statement:

- For a standard 2D convolution, let the input feature map have dimensions $H \times W \times C_{\text{in}}$, where H and W are height and width, and C_{in} is the number of input channels.
- The standard convolution applies C_{out} kernels of size $K \times K$, resulting in a complexity of:

$$H \times W \times C_{\text{in}} \times K^2 \times C_{\text{out}} \tag{1}$$

In contrast, for a depthwise separable convolution, the following hold:

1. **Depthwise convolution**:
   - Applies C_{in} kernels of size $K \times K$, one per channel.
   - Complexity:
$$H \times W \times C_{\text{in}} \times K^2 \tag{2}$$
2. **Pointwise convolution**:
   - Applies C_{out} 1x1 kernels, one for each channel combination.
   - Complexity:
$$H \times W \times C_{\text{in}} \times C_{\text{out}} \tag{3}$$

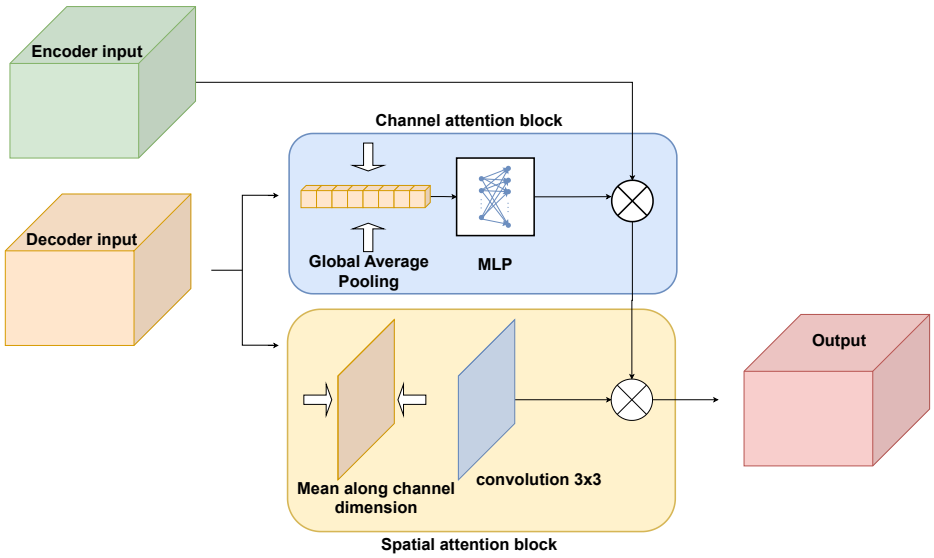


Fig. 4: The Visual Attention Block is depicted in more detail.

The combined complexity is:

$$H \times W \times C_{\text{in}} \times (K^2 + C_{\text{out}}) \tag{4}$$

This is significantly lower than the complexity of standard convolution, especially for large K and C_{in}.

The proposed DDM backbone encoder is illustrated in Fig. 7. It features three MultiRes blocks with depthwise separable convolutions and a dropout rate of 0.1, 0.1 and 0.2, respectively. They are followed by three dilated MultiRes blocks with depthwise separable convolutions and a dropout rate of 0.2, 0.3 and 0.3 respectively. The high dropout rate is employed to reduce overfitting, as was shown in our simulations.

The functionality of a MultiRes block is analyzed in more detail, as this is an essential part of our architecture. Figure 5 illustrates the structure.

The number of filters inside the block is calculated with the following

$$int(W * 0.167) + int(W * 0.333) + int(W * 0.5) \tag{5}$$

where $W = \alpha * f$ with α being the MultiRes block scaling factor and f being the number of filters if a non MultiRes block were used.

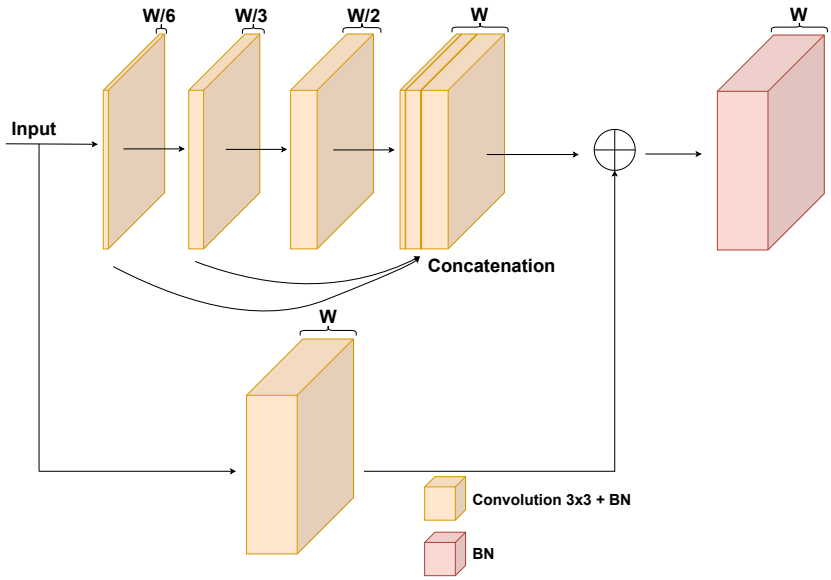


Fig. 5: The MultiRes block. $W = a \cdot f$ is the number of filters at the block output, where f is the number of filters at the block input

In more detail, the MultiRes block output Y is formulated as follows:

$$Y0 = Conv_{1x1}(X)$$

$$Y1 = Conv_{3x3}^{W/6}(X)$$
$$Y2 = Conv_{3x3}^{W/3}(Y1)$$
$$Y3 = Conv_{3x3}^{W/2}(Y2)$$
$$Y = ReLU(BN(Y0 + [Y1, Y2, Y3]))$$

where X is the block input and $[\cdot]$ is the concatenation function.

The primary goal of the MultiRes block is to capture image features at different scales without the overhead of larger convolutional kernels, such as 5×5 and 7×7. The cascading combination of 3×3 filters only achieves that with considerably less network size overhead.

The number of parameters in a convolutional block is defined as

$$\text{Parameters} = (k \times k \times C_{\text{in}} + 1) \times C_{\text{out}} \tag{6}$$

where:

- k = kernel size (i.e. k=5 for a 5×5),
- C_{in} = number of input channels,

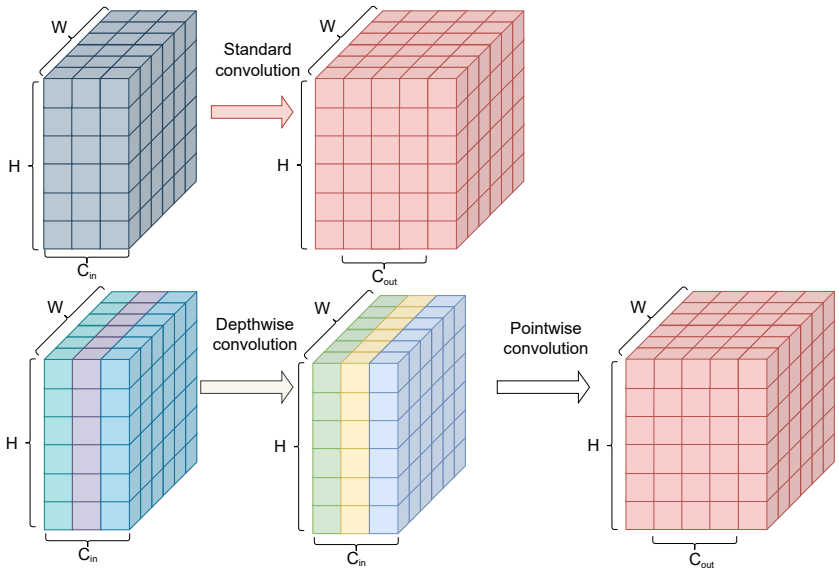


Fig. 6: The Depthwise separable convolution.

– C_{out} = number of output channels (filters),
– The +1 accounts for the bias term per filter.

An inception net module, with a 3×3, a 5×5 and a 7×7 kernel in parallel and with 16 input and 32 output filters would have the following list of parameters

Kernel	**Params**
3×3	4640
5×5	12832
7×7	25120
Total	42592

If, for simplification, the calculation of the number of filters of (5) is omitted, a MultiRes block would have the following number of parameters, which is about 1/3 of that of the larger kernel structure.

The proposed encoder is different compared to previous standard lightweight encoder, such as MobileNet [15, 18] , EfficientNet [16, 19] and ShuffleNet [20]. The difference is mainly the application for which these networks were designed. These networks were designed for image classification, whereas the proposed DDMnet is pre-trained for image classification, but is refined and designed to work for image instance segmentation and object detection. Therefore, the DDMnet was engineered

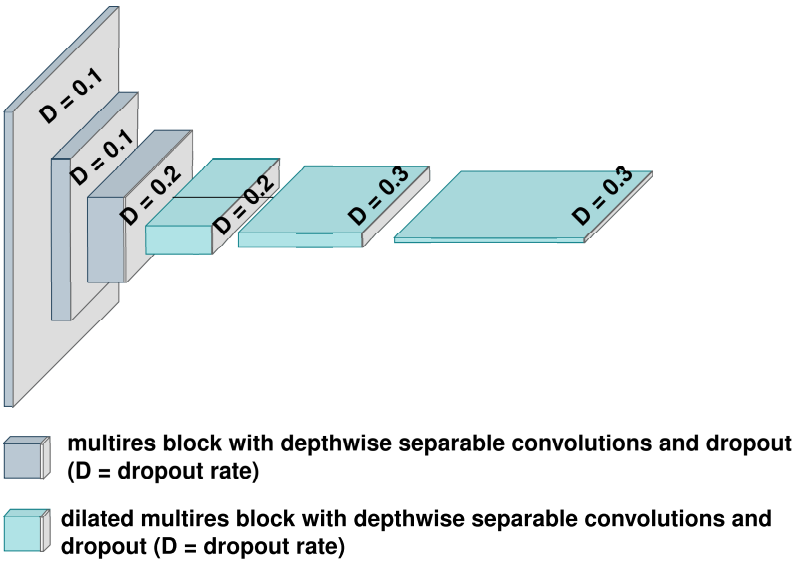


Fig. 7: The proposed DDM backbone encoder.

Kernel	Params
3×3	4640
3×3	4640
3×3	4640
1×1	544
Total	14464

to work efficiently for instance segmentation rather than being a generic image feature extraction network.

Overall, even though MultiRes blocks and Depthwise separable convolutions have appeared individually in prior works, our model integrates these architectural concepts, in a novel way, that is specifically tailored for instance segmentation problems. At the same time it focuses on real-time performance. Rather than simply using regular convolution blocks or stacking feature pyramids, through the use of MultiRes blocks, our model fuses multiscale features, a task that is essential for instance segmentation datasets, where class objects can appear in sizes with extensive variation. Depthwise separable convolutions preserve the representational power of convolutional blocks, allowing deeper feature maps at a reduced computational cost. Finally, the model is designed with compatibility to the Mask2Former decoder architecture in mind, taking advantage of the increased mask prediction stability of Mask2Former, compared to other standard backbones.

3 Ablation study

In our ablation study, several model characteristics is evaluated in order to establish the goal of a top performing yet minimal architecture. Architectural choices impact prediction accuracy and generalization ability, as well as the number of trainable parameters and computational complexity of models and for this reason, need to be carefully examined.

The Cityscapes dataset [21] is used for our experiments, where the model is trained for 200K iterations and batch size 16. The AdamW optimizer is also used with a polynomial decay learning rate of power 1, initialized at 0.0001. The optimizer also uses a weight decay of 0.05 for regularization. The models were trained with Python 3.8.10 using PyTorch 1.14.0 on a PC with i9-11900F, 64GB RAM, an NVidia RTX A6000 GPU 48GB, running Ubuntu Linux 22.04.

In addition to the above, all encoders were pretrained in the ImageNet-1K [22], with the same software and hardware configuration, for 90 epochs, with the SGD optimizer, initial learning rate 0.1, momentum 0.9, regularization weight decay 0.0001 and step learning rate reduction every 30 epochs with a factor of 0.1 (10%).

As a first step, in all our ablation study experiments where network architecture and size are examined, the performance of the Mask2Former architecture using a simple U-Net encoder is also listed, with 16 starting filters. This architecture is simply named "U-Net enc.". The common conclusion in all these cases is that the U-Net encoder results in a smaller network with limited performance, that leaves a lot of room for improvements while keeping the network size under real-time application constraints.

3.1 MultiRes block scaling coefficient

The MultiRes blocks use a scaling coefficient α to control the number of parameters and filters in the blocks, in order to benefit from factorizing larger convolution kernels with series of 3×3 convolutions. The authors decompose the number of filters W of the MultiRes blocks into the scaling factor and the corresponding number of standard U-Net layer filters U with the formula $W = \alpha \times U$.

In the following sections, references to the number of starting filters refer to the U-Net number U, in order to maintain a consistent reference to the baseline U-Net architecture properties.

The scaling factor proposed by the authors is 1.67 and this is considered to be our basis. We attempt to reduce the factor in order to assess the impact of this hyperparameter in producing compact but well-performing architectures. The results are in table 1. It is clear that the scale factor does influence performance and complexity but not to the extent that would lead to a notable trade-off.

3.2 Architecture size - Number of starting U-Net filters (U)

Our baseline architecture starts with 16 U-Net filters in the first layer and continues to increase their number as the feature maps are scaled down in size in the flow of the encoder. This results in 4M parameters, which is already small enough for platforms with restricted resources. The scale of the architecture can be increased by selecting

Method	Params	GFlops	Classification		Instance segmentation	
			Top-1 accuracy	Top-5 accuracy	AP	AP50
U-Net enc.	1.9M	0.55	58.13	79.65	28.06	50.01
$\alpha = 1.67$	4M	1.35	62.746	84.184	31.17	54.69
$\alpha = 1.43$	3.05M	0.87	61.366	82.508	30.09	52.97

Table 1: The table shows the effect of the MultiRes scale factor α in the model performance and complexity. The model is evaluated both in the ImageNet-1K classification pretraining and in the Cityscapes instance segmentation val dataset

more than 16 filters in the first layer. Experiments with 32 and 40 filters have been performed, in order to keep the architecture size at a small level (20M parameters). Table 2 shows the results. Clearly, the performance gains are not significant enough for the added complexity.

Method	Params	GFlops	Classification		Instance segmentation	
			Top-1 accuracy	Top-5 accuracy	AP	AP50
U-Net enc.	1.9M	0.55	58.13	79.65	28.06	50.01
$U = 16$	4M	1.35	62.75	84.18	31.17	54.69
$U = 32$	14M	-	71.92	90.8	36.6	62.75
$U = 40$	21.8M	-	72.48	90.9	37.1	64.1

Table 2: The table shows the effect of the starting U-Net filters U in the model performance and complexity. The model is evaluated both in the ImageNet-1K classification pretraining and in the Cityscapes instance segmentation val dataset. As expected, larger networks do improve performance but also significantly increase the network complexity, quickly pushing it close to or outside real-time boundaries

3.3 Depthwise separable convolutions

All convolutions have been replaced with depthwise separable convolutions, as described in Section 2.2. As seen in Table 4, the network size can be reduced significantly when depthwise separable convolutions are used instead of regular convolutions. If we just compare the network with 16 starting filters, it is about 2.5 times smaller when depthwise separable convolutions are used. This allows for the use of a larger number of starting filters, allowing for the construction of an encoder with higher representational capabilities. Section 2.2 describes in detail the architecture of a depthwise separable convolutional layer.

The model with 26 starting filters and depthwise separable convolutions is the best tradeoff in terms of the two contradicting objectives that we are after, performance and model size.

3.4 Loss function

The segmentation loss (mask loss) of the Mask2Former architecture is defined as

$$\mathcal{L} = \mathcal{L}_{mask} + \lambda_{cls}\mathcal{L}_{cls} \tag{7}$$

$\mathcal{L}_{cls}$ is the classification loss. $\mathcal{L}_{mask}$ is the mask loss and it is defined as

$$\mathcal{L}_{mask} = \lambda_{ce}\mathcal{L}_{ce} + \lambda_{dice}\mathcal{L}_{dice} \tag{8}$$

where $\mathcal{L}_{ce}$ is the binary cross-entropy loss (BCE) and $\mathcal{L}_{dice}$ is the Dice loss [23],[24]. The Dice loss is used to combat class imbalance in the training sets.

Alternatives to the binary cross-entropy loss for pixel label matching have been explored. The Mean Squared Error (MSE) and the Differentiable F-Measure (DFM) loss [25] have been evaluated. DFM loss is a differentiable version of the F-Measure as the loss function for problems with imbalanced datasets, which is given by the following:

$$\mathcal{L} = -\frac{(1+\beta^2)\sum_{i=1}^{N} y_i \cdot \hat{y}_i}{\sum_{i=1}^{N} \hat{y}_i + \beta^2 \cdot y_i} \tag{9}$$

where y_i is the label for sample i, $\hat{y}_i$ is the predicted value for that sample and β is the weighting factor between precision and recall of the regular F-Measure. In our experiments, $\beta = 1$ was used. The negative sign was employto change the monotonicity of the function, so that it can be used as a cost function. Table 3 shows that binary cross-entropy still outperforms the alternatives. The ablation study on loss functions has been performed over the $U = 26$ depthwise separable convolutions model of table 4.

Method	Top-1 accuracy	Top-5 accuracy	AP	AP50
BCE	**68.1**	**88.36**	**36.29**	**61.22**
MSE	62.38	78.95	28.01	55.03
DFM	64.86	81.02	29.54	58.6

Table 3: The table shows the effect of alternative loss functions for pixel label matching. Binary cross-entropy (BCE) performs better than Mean Squared Error (MSE) and Differentiable F-Measure (DFM)

Method	Params	GFlops	Classification		Instance segmentation	
			Top-1 accuracy	Top-5 accuracy	AP	AP50
U-Net enc.	1.9M	0.55	58.13	79.65	28.06	50.01
$U = 16$ (std convs)	4M	1.35	62.746	84.184	31.17	54.69
$U = 16$	1.67M	0.43	61.83	83.53	29.2	50.92
$U = 26$	**3.51M**	**1.02**	**68.1**	**88.36**	**36.29**	**61.22**
$U = 36$	5.96M	1.87	70.06	89.7	36.4	61.6
$U = 48$	9.72M	3.16	70.17	90.42	37.01	62.7
$U = 64$	16.09M	5.42	70.66	90.53	38.07	64.31

Table 4: The table shows the effect of the model size due to the starting U-Net filters U but when convolutions have been replaced with their depthwise separable counterparts. The network size is now considerably smaller for the same number of filters, which allows adding more filters while maintaining a low complexity. The highlighted version presents the best tradeoff between performance and size

4 Results

Our method targets the instance segmentation problem with the priority of achieving comparable performance while maintaining minimal model complexity.

The real-time capabilities of our method are exhibited in Table 5 through the GPU memory usage and batch/image inference times. Inference time of 7.8 ms per image results in a frame rate of 128 FPS for colored images of 1024×2048 resolution. Such a frame rate is far above the 30 FPS which can be generally considered as an upper threshold for human visual perception and real-time, interactive visualizations and videos. It is even higher than the 120 FPS, which is often used as the upper threshold for high-speed industrial applications. In addition to that, the GPU memory consumption of about 744 MiB per image, is favourable for compact low resource and embedded devices.

We first compare our model with competitor methods in terms of complexity. Table 6 shows that DDMnet is clearly dominant in terms of model complexity, far from any other competitor method.

Image size	Batch size	GPU Memory	Inference time
1024×2048	16	11900 MiB	0.125s

Table 5: DDMnet memory usage and inference time per batch of 16 images. This results in roughly 744 MiB GPU memory usage and 7.8ms inference time per image respectively, landing our method clearly in the real-time domain

Method	Parameters	GFLOPS
Mask2Former Resnet50 [6]	25M	-
Mask2Former Swin tiny [6]	28M	4.5
DDMnet	**3.5M**	**1.02**

Table 6: Model complexity comparison (measured on the ImageNet-1K classification dataset)

4.1 Cityscapes

Table 7 demonstrates how our method compares to the state-of-the-art models. Our method achieves high-quality performance with only a fraction of the complexity of the other methods. Being 10 to 100 times smaller, our network is still able to maintain comparable segmentation masking and separation quality results. In addition to that, the image table 8 shows that our method consistently exhibits smooth and non-noisy contours and accurate mask boundaries.

Method	Params	GFlops	AP	AP50
Mask2Former Resnet50 [6]	25M	144.7	37.4	61.9
Mask2Former Swin tiny [6]	28M	128	39.7	66.9
MP-Former Resnet50 [26]	25M	-	39.7	65.4
OpenSeed (T) [1]	>28M	-	38.2	-
OpenSeed (L) [1]	>197M	-	49.3	-
OneFormer [2]	219M	543	45.6	-
DiNAT-L [27]	220M	522	45.1	72.6
PointRend [28]	>25M	280	35.8	-
DDMnet	**3.51M**	103.6	36.29	61.22

Table 7: Instance segmentation on the Cityscapes dataset (the parameters refer to the backbone complexity

Furthermore, Table 8 lists the performance of the model on instance-level categories. Variations are mainly due to class imbalance. Still, variation is smaller compared to the ADE20K results that follow (Table 9), because the number of classes is smaller and categories, even though unbalanced, the classes are better represented, as compared to ADE20K.

4.2 ADE20K

The ADE20K [29] dataset contains densely annotated images covering a wide range of indoor and outdoor scenes. The dataset has 20,210 training images, 2,000 images for validation and 3,352 images for testing, classified over 150 object classes. Table

Class	AP	AP50
person	34,75	66,46
rider	27,95	64,26
car	58,85	84,96
truck	32,85	46,36
bus	59,05	74,16
train	37,85	55,86
motorcycle	18,05	44,16
bicycle	20,95	53,56
Average	**36.3**	**61.2**

Table 8: Instance-level ("Thing") categories for DDMnet on the Cityscapes dataset

10 shows that no architecture has a complexity small enough to be considered for real-time purposes.

In Table 9, the performance of our model for each instance-level category of the dataset is depicted. The variation among the different category results stems from the increased inherent class imbalance of the ADE20K dataset. As mentioned above, the number of classes is quite large and not all objects are represented equally in the training samples. In addition to that, object scale varies a lot and far too often phenomena, such as object occlusions appear in the training samples.

The ADE20K dataset is more demanding and challenging for the following reasons:

- Much larger number of classes
- Scene complexity with indoor and outdoor scenes with diverse objects, instead of the urban city street scenes of Cityscapes, where the variety of appearing objects is a lot smaller
- Diverse image resolutions, compared to the fixed Cityscapes image resolution
- Poorer image annotations, compared to the high-quality Cityscapes annotations

Unfortunately, the competing networks found in the literature feature 10 or 100 times more parameters than the proposed one. No lightweight architecture result was found on the ADE20K dataset. Nonetheless, our method still delivers descent performance and as seen in the sample images in Table 9 the results are visually acceptable. Accurate and smooth mask boundaries can be seen that align well with the detected objects without excessive overlap or gaps. In addition to that, a wide variation of object categories is detected, showing that our model generalizes well. Finally, good separation of instances belonging to the same object categories is also depicted.

5 Conclusions and future work

DDMnet is a compact and lightweight instance segmentation network with minimal architectural complexity that managed to achieve performance comparable to that of deep learning networks of very large size, as demonstrated by its benchmark results on the Cityscapes and ADE20K datasets. As a predominantly CNN-based model,

Fig. 8: Sample Instance segmentation results on the Cityscapes dataset. In the first image, it is clear that the boundary line at the bottom of the car is a lot smoother than the comparison images and closer to the ground truth. In the second image, the same trend is shown again, in the area under the mirror. These two examples, combined, show that our methods performs very well for both large and small scale object masks. Finally, the third image shows practically no noise in the annotated region, stressing the robustness of our method.

DDMnet underscores the continued importance of convolutional architectures in image segmentation tasks, highlighting their enduring relevance and potential for further advancements.

A known limitation of convolutional based feature extraction networks (backbone encoders) is the difficulty of harvesting global representations. Transformer based architectures can achieve much wider receptive fields at the cost of increased network size and complexity. Novelties that balance a better tradeoff between local and global feature extraction is a direction that can result in better yet more compact machine learning models and can serve as a promising future direction.

As a future direction, the main scope is to dive deeper into the exploration of both localized and global attention mechanisms for image segmentation tasks. By focusing on identifying and leveraging the most critical features, the aim is to optimize the network's ability to balance computational efficiency and performance.

For future work, models can become even lighter, without sacrificing performance by using techniques such as model pruning, quantization and better targeted attention mechanisms, tailored for segmentation tasks.

Category	AP	Category	AP	Category	AP
bed	52,31	windowpane	34,22	cabinet	35,40
person	23,79	door	23,29	table	26,56
curtain	49,80	chair	22,38	car	37,79
painting	54,29	sofa	43,97	shelf	8,64
mirror	42,92	armchair	29,18	seat	12,25
fence	5,41	desk	12,11	wardrobe	36,02
lamp	37,56	bathtub	59,51	railing	4,86
cushion	38,15	box	6,07	column	15,58
signboard	13,13	chest of drawers	40,39	counter	11,51
sink	36,03	fireplace	48,26	refrigerator	53,89
stairs	4,45	case	29,20	pool table	72,02
pillow	36,25	screen door	32,31	bookcase	10,13
coffee table	39,42	toilet	52,21	flower	13,47
book	15,99	bench	3,33	countertop	25,76
stove	43,41	palm	7,85	kitchen island	23,62
computer	11,36	swivel chair	19,83	boat	5,20
arcade machine	3,52	bus	23,00	towel	26,64
light	19,29	truck	9,01	chandelier	36,76
awning	13,27	streetlight	5,08	booth	8,02
television receiver	54,66	airplane	11,28	apparel	4,49
pole	9,83	bannister	2,62	ottoman	21,33
bottle	5,21	van	15,72	ship	3,28
fountain	1,87	washer	27,07	plaything	8,46
stool	14,75	barrel	10,38	basket	12,13
bag	2,39	minibike	8,14	oven	20,12
ball	3,97	food	2,18	step	7,93
trade name	12,74	microwave	51,47	pot	18,74
animal	10,78	bicycle	7,50	dishwasher	59,51
screen	48,70	sculpture	6,91	hood	48,05
sconce	16,48	vase	15,24	traffic light	5,05
tray	4,04	ashcan	19,28	fan	24,73
plate	11,80	monitor	3,62	bulletin board	16,28
radiator	28,36	glass	4,13	clock	20,74
flag	7,40				

Table 9: Instance-level ("Thing") categories for DDMnet on the ADE20K dataset

Furthermore, reducing the overall complexity of the model without compromising its segmentation accuracy paves the way for more efficient and effective solutions in real-time and resource-constrained environments.

Acknowledgements. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the RTX A6000 GPU used for this research.

Declarations

Ethical Approval

Not applicable.

Consent to Participate

Not applicable.

Consent to Publish

All authors give consent to Nature Springer to publish their details and the details of this research work.

Method	Params	AP	APm	APl
Mask2Former Resnet50 [6]	25M	26.4	28.9	43.1
Mask2Former Swin large [6]	215M	34.9	40.0	54.7
MP-Former Resnet50 [26]	25M	27.4	30.0	44.5
OpenSeed (T) [1]	>28M	35.1	-	-
OpenSeed (L) [1]	>197M	42.0	-	-
OneFormer [2]	219M	35.9	-	-
DiNAT-L [27]	220M	35.4	39.0	55.5
PointRend [28]	>25M	-	-	-
DDMnet	**3.51M**	21.95	23.22	36.87

Table 10: Instance segmentation on the ADE20K dataset

Fig. 9: Sample Instance segmentation results on the ADE20K dataset. The example images show detection, segmentation and instance separation performance with good object boundaries, considering the dataset particularities. As described, ADE20K is a more demanding and challenging dataset.

Data Availability Statement

All datasets used in this paper are publicly available at https://www.cityscapes-dataset.com and https://ade20k.csail.mit.edu.

Authors Contributions

Conceptualization, N.D., C.C., N.M. and I.P.; methodology, N.D. and N.M.; software, N.D. and C.C.; validation, N.D. and C.C.; formal analysis, N.D.; investigation, N.D.;

writing—original draft preparation, N.D.; writing—review and editing, N.M. and I.P.; visualization, N.D.; supervision, N.M. and I.P.

Funding

There was no funding for this research.

Competing Interests

There are no competing interests.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the RTX A6000 GPU used for this research.

References

[1] Zhang, H., Li, F., Zou, X., Liu, S., Li, C., Gao, J., Yang, J., Zhang, L.: A Simple Framework for Open-Vocabulary Segmentation and Detection (2023). https://arxiv.org/abs/2303.08131

[2] Jain, J., Li, J., Chiu, M.C., Hassani, A., Orlov, N., Shi, H.: Oneformer: One transformer to rule universal image segmentation. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2989–2998 (2022)

[3] Rossi, L., Karimi, A., Prati, A.: Self-balanced r-cnn for instance segmentation. Journal of Visual Communication and Image Representation **87**, 103595 (2022) https://doi.org/10.1016/j.jvcir.2022.103595

[4] Zhang, T., Xu, W., Luo, B., Wang, G.: Depth-Wise Convolutions in Vision Transformers for Efficient Training on Small Datasets (2024). https://arxiv.org/abs/2407.19394

[5] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., Girshick, R.: Segment Anything (2023). https://arxiv.org/abs/2304.02643

[6] Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention Mask Transformer for Universal Image Segmentation (2022). https://arxiv.org/abs/2112.01527

[7] Cheng, B., Schwing, A.G., Kirillov, A.: Per-Pixel Classification is Not All You Need for Semantic Segmentation (2021). https://arxiv.org/abs/2107.06278

[8] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. ArXiv **abs/1505.04597** (2015)

[9] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature Pyramid Networks for Object Detection (2017). https://arxiv.org/abs/1612.03144

[10] Detsikas, N., Mitianoudis, N., Papamarkos, N.: A dilated multires visual attention u-net for historical document image binarization. Signal Processing: Image Communication **122**, 117102 (2024) https://doi.org/10.1016/j.image.2024.117102

[11] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. arXiv (2016). https://doi.org/10.48550/ARXIV.1606.00915 . https://arxiv.org/abs/1606.00915

[12] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European Conference on Computer Vision (2018)

[13] Chen, L.-C., Papandreou, G., Schroff, F., Adam, H.: Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv (2017). https://doi.org/10.48550/ARXIV.1706.05587 . https://arxiv.org/abs/1706.05587

[14] Ibtehaz, N., Rahman, M.S.: MultiResUNet : Rethinking the u-net architecture for multimodal biomedical image segmentation. Neural Networks **121**, 74–87 (2020) https://doi.org/10.1016/j.neunet.2019.08.025

[15] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (2017). https://arxiv.org/abs/1704.04861

[16] Tan, M., Le, Q.V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (2020). https://arxiv.org/abs/1905.11946

[17] Chollet, F.: Xception: Deep Learning with Depthwise Separable Convolutions (2017). https://arxiv.org/abs/1610.02357

[18] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: MobileNetV2: Inverted Residuals and Linear Bottlenecks (2019). https://arxiv.org/abs/1801.04381

[19] Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. ArXiv **abs/1905.11946** (2019)

[20] Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6848–6856 (2017)

[21] Cordts, M., Omran, M., Ramos, S., Rehfel, T., Enzweiler, M., Benenson, R.,

Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

[22] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848

[23] Sudre, C.H., Li, W., Vercauteren, T.K.M., Ourselin, S., Cardoso, M.J.: Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. Deep learning in medical image analysis and multimodal learning for clinical decision support : Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, held in conjunction with MICCAI 2017 Quebec City, QC,... **2017**, 240–248 (2017)

[24] Milletari, F., Navab, N., Ahmadi, S.-A.: V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. arXiv (2016). https://doi.org/10.48550/ARXIV.1606.04797 . https://arxiv.org/abs/1606.04797

[25] Pastor-Pellicer, J., Zamora-Martínez, F., Boquera, S.E., Bleda, M.J.C.: F-measure as the error function to train neural networks. In: International Work-Conference on Artificial and Natural Neural Networks (2013). https://api.semanticscholar.org/CorpusID:657632

[26] Zhang, H., Li, F., Xu, H., Huang, S., Liu, S., Ni, L.M., Zhang, L.: MP-Former: Mask-Piloted Transformer for Image Segmentation (2023). https://arxiv.org/abs/2303.07336

[27] Hassani, A., Shi, H.: Dilated Neighborhood Attention Transformer (2023). https://arxiv.org/abs/2209.15001

[28] Kirillov, A., Wu, Y., He, K., Girshick, R.: PointRend: Image Segmentation as Rendering (2020). https://arxiv.org/abs/1912.08193

[29] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5122–5130 (2017). https://doi.org/10.1109/CVPR.2017.544