MiniSeg: An Extremely Minimum Network for Efficient COVID-19 Segmentation

Yu Qiu[1], **Yun Liu**[2*], **Shijie Li**[3], **Jing Xu**[1*]
[1] College of Artificial Intelligence, Nankai University
[2] College of Computer Science, Nankai University
[3] Department of Information Systems and Artificial Intelligence, University of Bonn
yqiu@mail.nankai.edu.cn, {vagrantlyun, jason.li.nku}@gmail.com, xujing@nankai.edu.cn

Abstract

The rapid spread of the new pandemic, *i.e.,* COVID-19, has severely threatened global health. Deep-learning-based computer-aided screening, *e.g.,* COVID-19 infected CT area segmentation, has attracted much attention. However, the publicly available COVID-19 training data are limited, easily causing overfitting for traditional deep learning methods that are usually data-hungry with millions of parameters. On the other hand, fast training/testing and low computational cost are also necessary for quick deployment and development of COVID-19 screening systems, but traditional deep learning methods are usually computationally intensive. To address the above problems, we propose MiniSeg, a lightweight deep learning model for efficient COVID-19 segmentation. Compared with traditional segmentation methods, MiniSeg has several significant strengths: i) it only has 83K parameters and is thus not easy to overfit; ii) it has high computational efficiency and is thus convenient for practical deployment; iii) it can be fast retrained by other users using their private COVID-19 data for further improving performance. In addition, we build a comprehensive COVID-19 segmentation benchmark for comparing MiniSeg to traditional methods.

Introduction

As one of the most severe pandemics in human history, *coronavirus disease 2019* (COVID-19) threatens global health with thousands of newly infected patients every day. Effective screening of infected patients is of high importance to the fight against COVID-19. The gold standard for COVID-19 diagnosis is the tried-and-true Reverse Transcription Polymerase Chain Reaction (RT-PCR) testing (Wang et al. 2020). Unfortunately, the sensitivity of RT-PCR testing is not high enough to prevent the spread of COVID-19 (Ai et al. 2020; Fang et al. 2020). Hence, computed tomography (CT) is used as a complementary tool for RT-PCR testing to improve screening sensitivity (Ai et al. 2020; Fang et al. 2020). Besides, CT analysis is necessary for clinical monitoring of disease severity (Inui et al. 2020). However, CT examination needs expert radiologists, but we severely lack experienced radiologists during this pandemic. Therefore, computer-aided systems are expected for automatic CT interpretation.

*Yun Liu and Jing Xu are corresponding authors.

When it comes to computer-aided COVID-19 screening, deep-learning-based technology is a good choice due to its uncountable successful stories (Sun et al. 2014; He et al. 2015; Liu et al. 2019, 2020, 2018a,b). However, directly applying traditional deep learning models for COVID-19 screening is suboptimal. On the one hand, these models usually have millions of parameters and thus require a large amount of labeled data for training. The problem is that the publicly available COVID-19 data are limited and thus easy to cause overfitting for traditional data-hungry models. On the other hand, traditional deep learning methods, especially the ones for image segmentation, are usually computationally intensive. Considering the current severe pandemic situation, fast training/testing and low computational load are essential for quick deployment and development of computer-aided COVID-19 screening systems.

It is a widely accepted concept that overfitting is easier to happen when a model has more parameters and less training data. To solve the above problems of COVID-19 segmentation, we observe that lightweight networks are not only uneasy to overfit owing to their small number of parameters but also likely to be efficient, making them suitable for computer-aided COVID-19 screening systems. Therefore, we think lightweight COVID-19 segmentation should be the technical solution of this paper. The key is to achieve accurate segmentation under the constraints of the number of network parameters and high efficiency. To achieve this goal, we find that the accuracy of image segmentation can be improved with effective multi-scale learning, which has significantly pushed forward the state of the arts of segmentation (Chen et al. 2018a, 2017, 2018b; Yang et al. 2018; Zhao et al. 2017; Mehta et al. 2018, 2019; Pohlen et al. 2017; Yu et al. 2018b). Hence, we resort to multi-scale learning to ensure the segmentation accuracy of lightweight networks.

With the above analyses, our effort starts with the design of an **A**ttentive **H**ierarchical **S**patial **P**yramid (**AHSP**) module for effective, lightweight multi-scale learning. AHSP first builds a spatial pyramid of dilated depthwise separable convolutions and feature pooling for learning multi-scale semantic features. Then, the learned multi-scale features are fused hierarchically to enhance the capacity of multi-scale representation. Finally, the multi-scale features are merged under the guidance of the attention mechanism, which learns to highlight essential information and filter out

noisy information in radiography images. With the AHSP module incorporated, we propose an extremely minimum network for efficient segmentation of COVID-19 infected areas in chest CT slices. Our model, namely **MiniSeg**, only has 83K parameters, two orders of magnitude less than traditional image segmentation methods, so that current limited COVID-19 data can be enough for training MiniSeg. At last, we build a comprehensive COVID-19 segmentation benchmark to compare MiniSeg to previous methods extensively. Experiments demonstrate that MiniSeg performs favorably against previous state-of-the-art segmentation methods with high efficiency, trained with limited COVID-19 data.

In summary, our contributions are threefold:

- We propose an **A**ttentive **H**ierarchical **S**patial **P**yramid (**AHSP**) module for effective, lightweight multi-scale learning that is essential for image segmentation.
- With AHSP incorporated, we present an extremely minimum network, **MiniSeg**, for accurate and efficient COVID-19 segmentation with limited training data.
- For an extensive comparison of MiniSeg with previous state-of-the-art segmentation methods, we build a comprehensive COVID-19 segmentation benchmark.

Related Work

Image segmentation is a hot topic due to its wide range of applications. Multi-scale learning plays an essential role in image segmentation because objects in images usually exhibit very large scale changes. Hence most current state-of-the-art methods aim at designing *fully convolutional networks* (FCNs) (Shelhamer, Long, and Darrell 2017) to learn effective multi-scale representations from input images. For example, U-Net (Ronneberger, Fischer, and Brox 2015), U-Net++ (Zhou et al. 2018), and Attention U-Net (Oktay et al. 2018) propose encoder-decoder architectures to fuse multi-scale deep features at multiple levels. DeepLab (Chen et al. 2018a) and its variants (Chen et al. 2017, 2018b; Yang et al. 2018) design ASPP modules using dilated convolutions with different dilation rates to learn multi-scale features. Besides the multi-scale learning, some studies focus on exploiting the global context information through pyramid pooling (Zhao et al. 2017), context encoding (Zhang et al. 2018a), or non-local operations (Huang et al. 2019; Zhu et al. 2019). The above models aim at improving segmentation accuracy without considering the model size and inference speed, so they are suboptimal for COVID-19 segmentation that only has limited training data and requires high efficiency.

Lightweight networks aim at reducing the parameters and improving the efficiency of deep networks. Convolutional factorization is an intuitive way to reduce the computational complexity of convolution operations. Specifically, many well-known network architectures decompose the standard convolution into multiple steps to reduce the computational complexity, including Flattened Model (Jin, Dundar, and Culurciello 2015), Inception networks (Szegedy et al. 2017), Xception (Chollet 2017), MobileNets (Howard et al. 2017; Sandler et al. 2018), and ShuffleNets (Zhang et al. 2018b; Ma et al. 2018). Among them, Xception and MobileNets factorize a convolution into a pointwise convolution and a depthwise separable convolution. ShuffleNets further factorize a pointwise convolution into a channel shuffle operation and a grouped pointwise convolution. There are also some studies focusing on efficient semantic segmentation network design (Wu et al. 2018; Mehta et al. 2018, 2019; Lo et al. 2019; Wang et al. 2019). Considering COVID-19 segmentation, our goal is to achieve higher accuracy and faster speed by enhancing multi-scale learning in a lightweight setting.

Computer-aided COVID-19 screening has attracted much attention to serve as a supplementary tool for RT-PCR testing to improve screening sensitivity. Some studies (Narin, Kaya, and Pamuk 2020; Gozes et al. 2020; Xu et al. 2020; Li et al. 2020a; Zhang et al. 2020; Wang and Wong 2020) design deep neural networks to classify chest X-rays or CT slices for COVID-19 screening. Fan et al. (2020) proposed a segmentation model for COVID-19 infected area segmentation from CT slices. However, their method also falls into the same category as previous segmentation methods and is thus suboptimal. Some public COVID-19 imaging datasets, such as COVID-19 X-ray Collection (Cohen et al. 2020), COVID-CT-Dataset (Zhao et al. 2020), COVID-19 CT Segmentation Dataset (Jenssen 2020), and COVID-19-CT-Seg (Jun et al. 2020), are introduced. In this paper, we focus on segmenting COVID-19 infected areas from chest CT slices.

Methodology

Attentive Hierarchical Spatial Pyramid Module

Although the factorization of a convolution operation into a pointwise convolution and a depthwise separable convolution (**DSConv**) can significantly reduce the number of network parameters and computational complexity, it usually comes with the degradation of accuracy (Howard et al. 2017; Sandler et al. 2018; Zhang et al. 2018b; Ma et al. 2018). Inspired by the fact that effective multi-scale learning plays an essential role in improving segmentation accuracy (Chen et al. 2018a, 2017, 2018b; Yang et al. 2018; Zhao et al. 2017; Mehta et al. 2018, 2019; Pohlen et al. 2017; Yu et al. 2018b), we propose the AHSP module for effective and efficient multi-scale learning in a lightweight setting. Besides some common convolution operations, such as vanilla convolution, pointwise convolution, and DSConv, we introduce the dilated DSConv convolution that adopts a dilated convolution kernel for each input channel. Suppose $\mathcal{F}_r^{k\times k}$ denotes a vanilla convolution, where $k \times k$ is the size of convolution kernel and r is the dilation rate. Suppose $\hat{\mathcal{F}}_r^{k\times k}$ denotes a depthwise separable convolution, where $k \times k$ and r have the same meaning as $\mathcal{F}_r^{k\times k}$. The subscript r will be omitted without ambiguity if we have a dilation rate of 1, *i.e.,* $r = 1$. For example, $\mathcal{F}^{1\times 1}$ represents a pointwise convolution (*i.e.,* 1×1 convolution). $\hat{\mathcal{F}}_2^{3\times 3}$ is a dilated 3×3 DSConv with a dilation rate of 2.

With the above definitions for basic operations, we continue by introducing the proposed AHSP module illustrated in Fig. 1. Let $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$ be the input feature map so that the output feature map is $\mathcal{E}(\mathbf{X}) \in \mathbb{R}^{C'\times H'\times W'}$, where $\mathcal{E}$ denotes the transformation function of AHSP for its input.

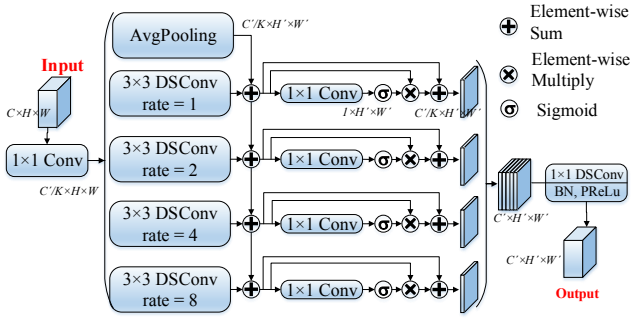

Figure 1: Illustration of the proposed AHSP module.

C, H, and W are the number of channels, height, and width of the input feature map $\mathbf{X}$, respectively. Similar definitions hold for C', H', and W'. The input feature map $\mathbf{X}$ is first processed by a pointwise convolution to shrink the number of channels into C'/K, in which K is the number of parallel branches which will be described later. This operation can be written as

$$\mathbf{S} = \mathcal{F}^{1\times1}(\mathbf{X}). \tag{1}$$

Then, the generated feature map $\mathbf{S}$ is fed into K parallel dilated DSConv, *i.e.,*

$$\mathbf{F}_k = \hat{\mathcal{F}}^{3\times3}_{2^{k-1}}(\mathbf{S}), \quad k = 1, 2, \cdots, K, \tag{2}$$

where the dilation rate is increased exponentially for enlarging the receptive field. Eqn. 2 is the basis for multi-scale learning with large dilation rates capturing large-scale contextual information and small dilation rates capturing local information. We also add an average pooling operation for $\mathbf{S}$ to enrich the multi-scale information, *i.e.,*

$$\mathbf{F}_0 = \mathrm{AvgPool}^{3\times3}(\mathbf{S}), \tag{3}$$

where $\mathrm{AvgPool}^{3\times3}$ represents the average pooling with a kernel size of 3×3. Note that we have $\mathbf{F}_k \in \mathbb{R}^{\frac{C'}{K}\times H'\times W'}$ for $k = 0, 1, \cdots, K$. If we have $H \neq H'$ or $W \neq W'$, the convolution and pooling operations in Eqn. 2 and Eqn. 3 will have a stride of 2 to downsample the feature map by a scale of 2; otherwise, the stride will be 1.

These multi-scale feature maps produced by Eqn. 2 and Eqn. 3 are aggregated in an attentive hierarchical manner. We first add them up hierarchically as

$$\begin{aligned} \dot{\mathbf{F}}_1 &= \mathbf{F}_0 + \mathbf{F}_1, \\ \dot{\mathbf{F}}_2 &= \dot{\mathbf{F}}_1 + \mathbf{F}_2, \\ &\cdots \\ \dot{\mathbf{F}}_K &= \dot{\mathbf{F}}_{K-1} + \mathbf{F}_K, \end{aligned} \tag{4}$$

where feature maps are gradually fused from small scales to large scales to enhance the representation capability of multi-scale learning. We further adopt a spatial attention mechanism to make the AHSP module automatically learn to focus on target structures of various scales. On the other hand, the attention mechanism can also learn to suppress irrelevant information at some feature scales and emphasize essential information at other scales. Such self-attention makes each scale speak for itself to decide how important it is in the multi-scale learning process. The transformation of $\dot{\mathbf{F}}$ by spatial attention can be formulated as

$$\ddot{\mathbf{F}}_k = \dot{\mathbf{F}}_k + \dot{\mathbf{F}}_k \otimes \sigma(\mathcal{F}^{1\times1}(\dot{\mathbf{F}}_k)), \quad k = 1, 2, \cdots, K, \tag{5}$$

in which σ is a *sigmoid* activation function and $\otimes$ indicates element-wise multiplication. The pointwise convolution in Eqn. 5 outputs a single-channel feature map which is then transformed to a spatial attention map by the *sigmoid* function. This attention map is replicated to the same size as $\dot{\mathbf{F}}_k$, *i.e.,* $\frac{C'}{K} \times H' \times W'$, before element-wise multiplication. Considering the efficiency, we can compute the attention map for all K branches together, like

$$\mathbf{A} = \sigma(\mathcal{F}^{1\times1}(\mathrm{Concat}(\dot{\mathbf{F}}_1, \dot{\mathbf{F}}_2, \cdots, \dot{\mathbf{F}}_K))), \tag{6}$$

where $\mathrm{Concat}(\cdot)$ means to concatenate a series of feature maps along the channel dimension. The pointwise convolution in Eqn. 6 is a K-grouped convolution with K output channels, so we have $\mathbf{A} \in \mathbb{R}^{K\times H'\times W'}$. Hence, we can rewrite Eqn. 5 as

$$\ddot{\mathbf{F}}_k = \dot{\mathbf{F}}_k + \dot{\mathbf{F}}_k \otimes \mathbf{A}[k], \quad k = 1, 2, \cdots, K, \tag{7}$$

in which $\mathbf{A}[k]$ means the k-th channel of $\mathbf{A}$.

Finally, we merge and fuse the above hierarchical feature maps as

$$\begin{aligned} \ddot{\mathbf{F}} &= \mathrm{Concat}(\ddot{\mathbf{F}}_1, \ddot{\mathbf{F}}_2, \cdots, \ddot{\mathbf{F}}_K), \\ \mathcal{E}(\mathbf{X}) &= \mathrm{PReLU}(\mathrm{BatchNorm}(\mathcal{F}^{1\times1}(\ddot{\mathbf{F}}))), \end{aligned} \tag{8}$$

where $\mathrm{BatchNorm}(\cdot)$ denotes the batch normalization (Ioffe and Szegedy 2015) and $\mathrm{PReLU}(\cdot)$ indicates PReLU (*i.e.,* Parametric ReLU) activation function (He et al. 2015). The pointwise convolution in Eqn. 8 is a K-grouped convolution with C' output channels, so that this pointwise convolution aims at fusing $\ddot{\mathbf{F}}_k$ ($k = 1, 2, \cdots, K$) separately, *i.e.,* adding connection to channels for depthwise convolutions in Eqn. 2. The fusion among various feature scales is achieved through the first pointwise convolution (*i.e.,* Eqn. 1) in the subsequent AHSP module of MiniSeg and the hierarchical feature aggregation (*i.e.,* Eqn. 4). Such a design can reduce the number of convolution parameters in Eqn. 8 by K times when compared with that using a vanilla pointwise convolution, *i.e.,* C'^2/K *vs.* C'^2.

Given an input feature map $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, we can compute the output feature map $\mathcal{E}(\mathbf{X}) \in \mathbb{R}^{C'\times H'\times W'}$ of an AHSP module using Eqn. 1 - Eqn. 8. We can easily find that increasing K will reduce the number of AHSP parameters. Considering the balance between segmentation accuracy and efficiency, we set $K = 4$ in our experiments. The proposed AHSP module not only significantly reduces the number of parameters but also enables us to learn effective multi-scale features so that we can adopt the limited COVID-19 data to train a high-quality segmenter.

Network Architecture

MiniSeg has an encoder-decoder structure. The encoder sub-network focuses on learning effective multi-scale representations for the input image. The decoder sub-network gradually aggregates the representations at different levels of the

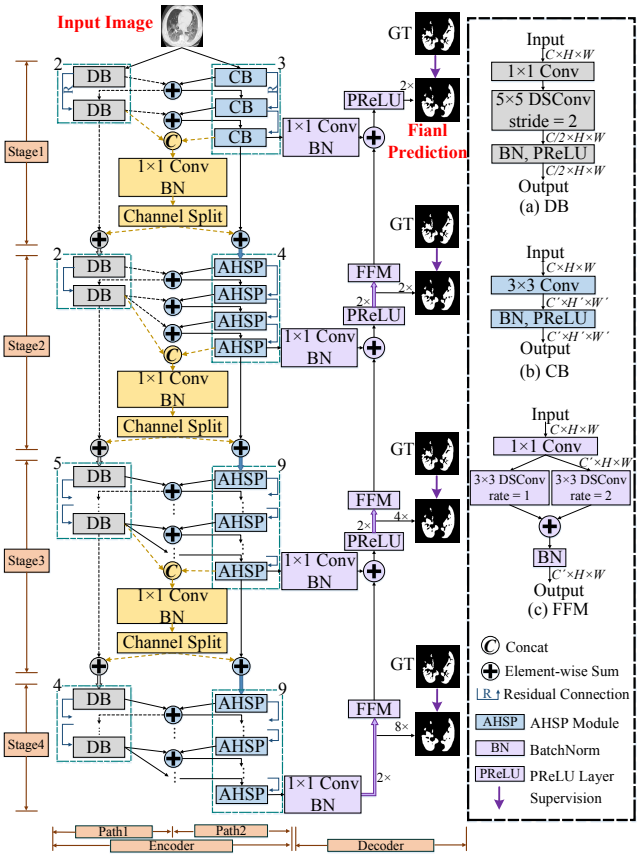

Figure 2: Network architecture of the proposed MiniSeg.

encoder to predict COVID-19 infected areas. The network architecture of MiniSeg is displayed in Fig. 2.

The encoder sub-network uses AHSP as the basic module, consisting of two paths connected through a series of nested skip pathways. Suppose $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ denotes an input chest CT slice, where a grayscale CT slice is replicated three times to make its number of channels the same as color images. The input $\mathbf{I}$ is downsampled four times, resulting in four scales of $1/2$, $1/4$, $1/8$, and $1/16$, with four stages processing such four scales, respectively. Downsampling happens in the first block of each stage.

Suppose in the encoder sub-network we denote the output feature map of the i-th stage and the j-th block as $\mathbf{E}_j^i$, *w.r.t.* $i \in \{1,2,3,4\}$ and $j \in \{1,2,\cdots,N_i\}$, where N_i indicates the number of blocks at the i-th stage. Therefore, we have $\mathbf{E}_j^i \in \mathbb{R}^{C_i\times\frac{H}{2^i}\times\frac{W}{2^i}}$, in which C_i is the number of feature channels at the i-th stage. The abovementioned block refers to the proposed AHSP module except for the first stage whose basic block is the vanilla **Convolution Block** (**CB**). Since the number of feature channels at the first stage (*i.e.,* C_1) is small, the vanilla convolution will not introduce too many parameters. Without ambiguity, let $\mathcal{E}_j^i(\cdot)$ be the transformation function of the i-th stage and the j-th block without distinguishing whether this block is a vanilla convolution or an AHSP module. For the another path, we propose a **Downsampler Block** (**DB**). The transformation function of a DB block is denoted as $\mathcal{Q}_k^i(\cdot)$, *w.r.t.* $i \in \{1,2,3,4\}$ and $k \in \{1,2,\cdots,M_i\}$, where M_i denotes the number of DB at the i-th stage. We define DB as

$$\mathcal{Q}_k^i(\mathbf{X}) = \mathrm{PReLU}(\mathrm{BatchNorm}(\hat{\mathcal{F}}^{5\times5}(\mathcal{F}^{1\times1}(\mathbf{X})))), \quad (9)$$

where $\hat{\mathcal{F}}^{5\times5}(\cdot)$ has a stride of 2 for downsampling when we have $k = 1$. Suppose the output of $\mathcal{Q}_k^i(\cdot)$ is $\mathbf{Q}_k^i$.

Therefore, for the first block of the first stage, we have

$$\mathbf{E}_1^1 = \mathcal{E}_1^1(\mathbf{I}), \quad \mathbf{Q}_1^1 = \mathcal{Q}_1^1(\mathbf{I}). \quad (10)$$

For the first block of other stages, we compute the output feature map as

$$\begin{aligned}\mathbf{E}^{i-1} &= \mathcal{F}^{1\times1}(\mathrm{Concat}(\mathbf{E}_{N_{i-1}}^{i-1}, \mathbf{Q}_{M_{i-1}}^{i-1})),\\ \mathbf{E}_1^i &= \mathcal{E}_1^i(\mathrm{Split}(\mathbf{E}^{i-1}) + \mathbf{E}_{N_{i-1}}^{i-1}),\\ \mathbf{Q}_1^i &= \mathcal{Q}_1^i(\mathrm{Split}(\mathbf{E}^{i-1}) + \mathbf{Q}_{M_{i-1}}^{i-1}),\end{aligned} \quad (11)$$

where we have $i \in \{2,3,4\}$. The operation $\mathrm{Split}(\cdot)$ is to split a feature map along the channel dimension into two chunks, which are fed into $\mathcal{E}_1^i$ and $\mathcal{Q}_1^i$, respectively. Here, $\mathcal{E}_1^i(\cdot)$ and $\mathcal{Q}_1^i(\cdot)$ ($i \in \{1,2,3,4\}$) have a stride of 2 for downsampling. Instead of only using on-the-fly element-wise sum (Eqn. 12 and Eqn. 13), through Eqn. 11, we conduct a "concat-fuse-split" operation to fully integrate the features from the two paths, as concatenation can do better for feature fusion than sum by avoiding the information loss of sum (Huang et al. 2017). $\mathrm{Split}(\cdot)$ is used to handle the increased number of channels brought by concatenation.

For other blocks, the output feature map is

$$\begin{aligned}\mathbf{E}_j^i &= \mathcal{E}_j^i(\mathbf{E}_{j-1}^i + \mathbf{Q}_{j'}^i) + \mathbf{E}_{j-1}^i,\\ &\textit{w.r.t.}\ i \in \{1,2,3,4\} \text{ and } j \in \{2,3,\cdots,N_i\},\end{aligned} \quad (12)$$

where $\mathcal{E}_j^i(\cdot)$ has a stride of 1 and a residual connection is included for better optimization. We have $j' = j-1$ if we also have $j-1 \leq M_i$; otherwise, we have $j' = M_i$. The computation of $\mathbf{Q}_k^i$ can be formulated as

$$\begin{aligned}\mathbf{Q}_k^i &= \mathcal{Q}_k^i(\mathbf{Q}_{k-1}^i + \mathbf{E}_{k-1}^i) + \mathbf{Q}_{k-1}^i,\\ &\textit{w.r.t.}\ i \in \{1,2,3,4\} \text{ and } k \in \{2,3,\cdots,M_i\}.\end{aligned} \quad (13)$$

Through Eqn. 12 and Eqn. 13, the two paths of the encoder sub-network build nested skip connections. Such a design benefits the multi-scale learning of the encoder. Considering the balance among the number of network parameters, segmentation accuracy, and efficiency, we set C_i to $\{8,24,32,64\}$, N_i to $\{3,4,9,9\}$, and M_i to $\{2,2,5,4\}$ for $i \in \{1,2,3,4\}$, respectively.

The decoder sub-network is simple for efficient multi-scale feature decoding. Since the top feature map of the encoder has a scale of $1/16$ of the original input, it is sub-optimal to predict COVID-19 infected areas directly due to the loss of fine details. Instead, we utilize a simple decoder sub-network to gradually upsample and fuse the learned feature map at each scale. A **Feature Fusion Module** (**FFM**)

Datasets	#Total/#COVID	#Patients
COVID-19-CT100 (Jenssen 2020)	100/100	∼60
COVID-19-P9 (Jenssen 2020)	829/373	9
COVID-19-P20 (Jun et al. 2020)	1844/1844	20
COVID-19-P1110 (Morozov et al. 2020)	785/785	50

Table 1: A summary of public COVID-19 CT datasets. #Total and #COVID denote the numbers of all or COVID-19 infected CT slices, respectively.

is proposed for feature aggregation. Let $\mathcal{D}_i(\cdot)$ represent the function of FFM:

$$\begin{aligned} \mathbf{S}'_i &= \mathcal{F}^{1\times1}(\mathbf{X}), \\ \mathcal{D}_i(\mathbf{X}) &= \mathrm{BatchNorm}(\hat{\mathcal{F}}^{3\times3}(\mathbf{S}'_i) + \hat{\mathcal{F}}_2^{3\times3}(\mathbf{S}'_i)), \end{aligned} \tag{14}$$

in which $\mathcal{D}_i(\mathbf{X})$ $(i = 1, 2, 3)$ has C_i channels as the pointwise convolution is utilized to adjust such number of channels. We denote the feature map in the decoder as $\mathbf{D}_i \in \mathbb{R}^{C_i \times \frac{H}{2^i} \times \frac{W}{2^i}}$, and we have $\mathbf{D}_4 = \mathrm{BatchNorm}(\mathcal{F}^{1\times1}(\mathbf{E}^4_{N_4}))$. We compute $\mathbf{D}_i$ $(i = 3, 2, 1)$ as

$$\begin{aligned} \mathbf{S}''_i &= \mathcal{D}_i(\mathrm{Upsample}(\mathbf{D}_{i+1}, 2)), \\ \mathbf{D}_i &= \mathrm{PReLU}(\mathbf{S}''_i + \mathrm{BatchNorm}(\mathcal{F}^{1\times1}(\mathbf{E}^i_{N_i}))), \end{aligned} \tag{15}$$

where $\mathrm{Upsample}(\cdot, t)$ means to upsample a feature map by a scale of t using bilinear interpolation. In this way, the decoder sub-network enhances the high-level semantic features with low-level fine details, so that MiniSeg can make accurate predictions for COVID-19 infected areas.

With $\mathbf{D}_i$ $(i = 1, 2, 3, 4)$ computed, we can make dense prediction using a pointwise convolution, *i.e.,*

$$\mathbf{P}_i = \mathrm{Softmax}(\mathrm{Upsample}(\mathcal{F}^{1\times1}(\mathbf{D}_i), 2^i)), \tag{16}$$

where $\mathrm{Softmax}(\cdot)$ is the standard *softmax* function and this pointwise convolution has two output channels representing two classes of background and COVID-19, respectively. $\mathbf{P}_i \in \mathbb{R}^{H\times W}$ is the predicted class label map. We utilize $\mathbf{P}_1$ as the final output prediction. In training, we impose deep supervision (Lee et al. 2015) by replacing the *softmax* function in Eqn. 16 with the standard cross-entropy loss function.

Experiments

Experimental Setup

Implementation details. We implement the proposed MiniSeg network using the well-known PyTorch framework (Paszke et al. 2017). Adam optimization (Kingma and Ba 2015) is used for training with the weight decay of 1e-4. We adopt the learning rate policy of *poly*, where the initial learning rate is 1e-3. We train 80 epochs on the training set with a batch size of 5. We train all previous state-of-the-art segmentation methods using the same training settings as MiniSeg for a fair comparison.

Dataset. We utilize four open-access COVID-19 CT segmentation datasets, *i.e.,* two sub-datasets from *COVID-19 CT Segmentation Dataset* (Jenssen 2020), *COVID-19 CT Lung and Infection Segmentation Dataset* (Jun et al. 2020), and *MosMedData* (Morozov et al. 2020), to evaluate MiniSeg. According to the number of CT slices or the number of COVID-19 patients, we rename these datasets as COVID-19-CT100, COVID-19-P9, COVID-19-P20, and COVID-19-P1110 for convenience, respectively. The information of these datasets is summarized in Tab. 1. We utilize the standard cropping and random flipping for data augmentation for MiniSeg and all baselines in training. Moreover, we perform **5-fold cross-validation** to avoid statistically significant differences in performance evaluation.

SB	MB	AH	TP	CS	Metrics (%)				
					mIoU	SEN	SPE	DSC	HD
✔					75.78	71.80	96.78	59.63	92.12
	✔				76.31	76.22	97.40	61.57	88.05
	✔	✔			76.58	77.89	97.61	62.06	83.71
	✔	✔	✔		76.66	78.72	97.02	62.05	78.67
	✔	✔	✔	✔	**78.33**	**79.62**	**97.71**	**64.84**	**71.69**

Table 2: Effect of the main components in MiniSeg on the COVID-19-P1110 dataset. Note that the metric HD does not have the unit of %.

PReLU	DE	DS	CB	DB 5×5	FFM	Metrics (%)				
						mIoU	SEN	SPE	DSC	HD
ReLU						76.92	75.41	96.90	62.11	78.39
	✗					73.11	76.31	97.45	55.35	76.72
		✗				76.45	**80.94**	96.38	62.46	87.27
			AHSP			76.71	78.38	96.38	62.05	78.99
				3×3		76.54	78.43	97.09	61.81	80.61
					AHSP	77.15	78.69	97.33	62.46	82.98
						78.33	79.62	**97.71**	**64.84**	**71.69**

Table 3: Effect of some design choices on COVID-19-P1110. Each design choice is replaced with the operation in the table or directly removed (✗). DE: Decoder. DS: Deep Supervision.

Evaluation metrics. We evaluate COVID-19 segmentation accuracy using five popular evaluation metrics in medical imaging analysis, *i.e.,* mean intersection over union (mIoU), sensitivity (SEN), specificity (SPC), Dice similarity coefficient (DSC), and Hausdorff distance (HD). Specifically, mIoU, SEN, SPC, and DSC range between 0 and 1. The larger these values, the better the model. Note that a lower value of HD indicates better segmentation accuracy. Moreover, we also report the number of parameters, the number of FLOPs, and speed, tested using a 512×512 input image and a TITAN RTX GPU.

Ablation Studies

Effect of main components. As shown in Tab. 2, we start with a **single-branch (SB)** module that only has the DSConv with a dilation rate of 1. We replace all AHSP modules in MiniSeg with such SB modules and remove the two-path design of the MiniSeg encoder (the 1st line of Tab. 2). Then, we extend such an SB module to a **multi-branch (MB)** module using the spatial pyramid as in the AHSP module to demonstrates the importance of multi-scale learning (the 2nd line of Tab. 2). Next, we add the **attentive hierarchical fusion strategy (AH)** to get the AHSP module to proves the superiority of the attentive hierarchical fusion (the 3rd line of Tab. 2). We continue by adding the **two-path design (TP)** to the encoder sub-network to validates that such a two-path design can benefit the network optimization (the 4th line of

Method	Backbone	ImageNet	#Param	FLOPs	Speed
U-Net	-	No	8.43M	65.73G	57.3fps
FCN-8s	VGG16	Yes	15.53M	105.97G	4.5fps
SegNet	-	No	28.75M	160.44G	3.0fps
FRRN	-	No	17.30M	237.70G	15.8fps
PSPNet	ResNet50	Yes	64.03M	257.79G	17.1fps
DeepLabv3	ResNet50	Yes	38.71M	163.83G	25.3fps
DenseASPP	-	No	27.93M	122.28G	19.3fps
DFN	ResNet50	Yes	43.53M	81.88G	56.2fps
EncNet	ResNet50	Yes	51.25M	217.46G	18.1fps
DeepLabv3+	Xception	Yes	53.33M	82.87G	3.4fps
BiSeNet	ResNet18	Yes	12.50M	13.01G	172.4fps
UNet++	-	No	8.95M	138.37G	26.8fps
Attention U-Net	-	No	8.52M	67.14G	49.2fps
OCNet	ResNet50	Yes	51.60M	220.69G	19.3fps
DUpsampling	ResNet50	Yes	28.46M	123.01G	36.5fps
DANet	ResNet50	Yes	64.87M	275.72G	16.4fps
CCNet	ResNet50	Yes	46.32M	197.92G	40.0fps
ANNNet	ResNet50	Yes	47.42M	203.07G	32.8fps
GFF	ResNet50	Yes	90.57M	374.03G	17.5fps
Inf-Net	ResNet50	Yes	30.19M	27.30G	155.9fps
MobileNet	MobileNet	Yes	3.13M	3.02G	416.7fps
MobileNetv2	MobileNetv2	Yes	2.17M	1.60G	137.0fps
ShuffleNet	ShuffleNet	Yes	0.92M	0.75G	116.3fps
ShuffleNetv2	ShuffleNetv2	Yes	1.22M	0.77G	142.9fps
EfficientNet	EfficientNet	No	8.37M	13.19G	48.1fps
ENet	-	No	0.36M	1.92G	71.4fps
ESPNet	-	No	0.35M	1.76G	125.0fps
CGNet	-	No	0.49M	3.40G	73.0fps
ESPNetv2	-	No	0.34M	0.77G	73.0fps
EDANet	-	No	0.68M	4.43G	147.1fps
LEDNet	-	No	2.26M	6.32G	94.3fps
MiniSeg	-	No	**82.91K**	**0.50G**	**516.3fps**

Table 4: Comparison of MiniSeg to previous state-of-the-art methods in terms of parameters, FLOPs, and speed.

Tab. 2). At last, we add the **channel split (CS)** operation to obtain the final MiniSeg model (the 5^{th} line of Tab. 2). These ablation studies demonstrate that the main components in MiniSeg are all effective for COVID-19 segmentation.

Effect of some design choices. We continue by evaluating the design choices of MiniSeg. The results are provided in Tab. 3. First, we replace the PReLU activation function with the ReLU function. Second, we remove the decoder sub-network and change the stride of the last stage from 2 to 1, so we can directly make predictions at the scale of $1/8$ and upsample to the original size, as in previous studies (Mehta et al. 2018; Lo et al. 2019; Wu et al. 2018; Paszke et al. 2016; Chen et al. 2018a). Third, we remove deep supervision in training. Fourth, we replace Convolution Blocks (CB) in the first stage with AHSP modules. Fifth, we replace the 5×5 DSConv in the Downsampler Blocks (DB) with 3×3 DSConv. Sixth, we replace the Feature Fusion Modules (FFM) in the decoder sub-network with AHSP modules. The default setting achieves the best overall performance, demonstrating the effectiveness of our designs.

Comparison with State-of-the-art Methods

Quantitative Evaluation. To compare MiniSeg to previous state-of-the-art competitors and promote COVID-19 segmentation research, we build a comprehensive benchmark. This benchmark contains 31 previous state-of-the-art image segmentation methods, including U-Net (Ronneberger, Fischer, and Brox 2015), FCN-8s (Shelhamer, Long, and Darrell 2017), SegNet (Badrinarayanan, Kendall, and Cipolla 2017), FRRN (Pohlen et al. 2017), PSPNet (Zhao et al. 2017), DeepLabv3 (Chen et al. 2017), DenseASPP (Yang et al. 2018), DFN (Yu et al. 2018b), EncNet (Zhang et al. 2018a), DeepLabv3+ (Chen et al. 2018b), BiSeNet (Yu et al. 2018a), UNet++ (Zhou et al. 2018), Attention U-Net (Oktay et al. 2018), OCNet (Yuan and Wang 2018), DUpsampling (Tian et al. 2019), DANet (Fu et al. 2019), CCNet (Huang et al. 2019), ANNNet (Zhu et al. 2019), GFF (Li et al. 2020b), Inf-Net (Fan et al. 2020), MobileNet (Howard et al. 2017), MobileNetv2 (Sandler et al. 2018), ShuffleNet (Zhang et al. 2018b), ShuffleNetv2 (Ma et al. 2018), EfficientNet (Tan and Le 2019), ENet (Paszke et al. 2016), ESPNet (Mehta et al. 2018), CGNet (Wu et al. 2018), ESPNetv2 (Mehta et al. 2019), EDANet (Lo et al. 2019), and LEDNet (Wang et al. 2019). Among them, Inf-Net is designed for COVID-19 segmentation. MobileNet, MobileNetv2, ShuffleNet, ShuffleNetv2, and EfficientNet are designed for lightweight image classification. We view them as the encoder and add the decoder of MiniSeg to them so that they are reformed as image segmentation models. ENet, ESPNet, CGNet, ESPNetv2, EDANet, and LEDNet are well-known lightweight segmentation models. The code of these methods is provided online by the authors. We believe that this benchmark would be useful for future research on COVID-19 segmentation.

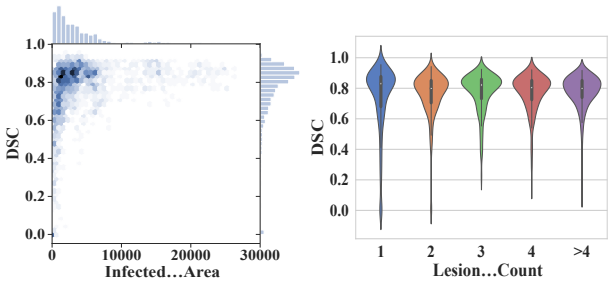

Figure 3: Statistical analysis for MiniSeg on COVID-19-P20. (a) The DSC score *vs.* the infected area; (b) The DSC score *vs.* the lesion count in the corresponding CT slice.

The comparison between MiniSeg and competitors, in terms of the number of parameters, the number of FLOPs, and speed, is summarized in Tab. 4. We can clearly see that the numbers of parameters and FLOPs of MiniSeg are extremely small. Meanwhile, the speed of MiniSeg is much faster than others. The numerical evaluation results of MiniSeg and other competitors are presented in Tab. 5. MiniSeg consistently achieves the best or close to the best performance in terms of all metrics on all datasets. For the metric of SPC, MiniSeg performs slightly worse than the best method on COVID-19-CT100 and COVID-19-P9. On the COVID-19-P1110 dataset, MiniSeg does not achieve the best results in terms of SEN. The fact that MiniSeg consistently outperforms other competitors demonstrates its effectiveness and superiority in COVID-19 infected area segmentation. Note that MiniSeg does not need to be pretrained on ImageNet (Russakovsky et al. 2015) owing to its small model size. Therefore, we can come to the conclusion that MiniSeg has a low computational load, a fast speed, and good accuracy, making it convenient for practical deploy-

Methods	COVID-19-CT100					COVID-19-P9					COVID-19-P20					COVID-19-P1110				
	mIoU	SEN	SPC	DSC	HD	mIoU	SEN	SPC	DSC	HD	mIoU	SEN	SPC	DSC	HD	mIoU	SEN	SPC	DSC	HD
U-Net	77.56	72.24	97.71	68.37	94.25	76.51	88.53	98.93	65.69	133.64	81.81	82.73	97.92	72.66	61.66	74.26	**81.85**	97.33	58.62	95.72
FCN-8s	71.85	66.47	93.56	58.11	104.68	81.20	87.12	98.40	72.67	91.32	82.54	84.10	98.02	73.60	51.47	70.51	80.75	97.08	53.33	84.43
SegNet	75.02	80.02	96.34	64.84	109.05	73.88	73.59	98.79	62.07	98.38	79.55	81.68	98.44	69.68	77.28	72.32	76.77	97.24	55.92	105.42
FRRN	79.20	78.47	97.50	71.27	86.56	80.83	86.26	99.54	74.03	84.34	80.61	80.75	97.53	71.43	61.28	73.84	75.45	95.80	58.86	87.11
PSPNet	75.61	70.82	96.47	64.55	99.76	82.15	86.84	99.19	74.85	94.40	81.60	83.44	98.17	71.60	65.60	71.41	80.34	97.40	54.82	87.06
DeepLabv3	81.30	84.80	97.48	74.65	81.77	81.50	85.23	98.56	73.10	95.72	80.26	81.60	97.78	70.96	60.50	72.91	80.45	96.85	55.70	81.35
DenseASPP	78.43	81.14	97.02	70.37	156.23	72.78	70.26	98.65	65.53	98.61	81.11	82.21	97.80	71.68	64.05	74.84	69.38	95.65	57.24	76.61
DFN	81.07	84.27	97.49	74.45	83.73	79.19	85.78	98.64	69.93	106.23	79.13	80.96	96.51	69.46	66.56	73.40	80.12	97.13	57.31	87.10
EncNet	71.28	74.11	95.20	62.83	119.55	81.35	86.88	98.65	72.62	94.77	82.43	84.94	98.03	71.60	71.57	71.65	81.23	96.65	54.89	77.82
DeepLabv3+	79.45	79.58	97.55	71.70	93.09	81.29	77.93	99.30	73.48	81.95	81.26	81.61	95.35	42.79	182.14	74.14	74.65	97.26	57.16	102.78
BiSeNet	63.09	74.07	87.41	58.66	110.47	72.33	67.17	96.35	55.40	164.07	78.08	76.13	97.07	65.24	85.94	70.29	70.90	95.49	52.26	95.11
UNet++	77.64	77.26	97.28	69.04	91.73	77.95	86.83	99.39	69.27	104.83	80.73	79.61	96.75	70.34	63.01	73.39	75.67	96.13	59.08	88.21
Attention U-Net	77.71	74.75	97.56	68.93	92.15	76.26	76.39	99.24	66.74	102.43	80.70	82.92	97.41	71.27	64.91	74.62	81.32	97.63	59.34	95.16
OCNet	69.29	72.86	89.38	56.14	105.66	81.14	87.41	98.71	72.94	113.21	80.74	80.71	95.82	69.36	56.60	72.05	79.67	97.64	53.97	97.38
DUpsampling	81.69	84.54	97.60	75.27	81.07	79.96	74.42	96.38	69.60	64.62	81.05	79.37	96.34	71.01	60.19	72.16	65.18	91.77	53.98	72.29
DANet	73.57	66.30	92.76	61.34	99.11	81.59	88.78	99.13	73.82	114.69	78.35	79.87	97.31	67.04	83.13	73.47	75.00	95.80	56.07	74.04
CCNet	75.24	69.55	95.92	63.99	98.03	81.27	86.61	99.16	73.93	90.84	82.22	82.93	97.76	73.13	56.98	72.02	79.16	96.29	54.83	83.07
ANNNet	73.93	66.73	95.72	62.06	102.43	79.52	85.20	98.35	69.55	109.31	81.92	84.10	98.13	72.72	56.99	72.28	81.19	97.30	55.21	83.16
GFF	75.75	69.80	97.53	63.88	103.87	81.20	85.35	98.46	72.61	113.48	82.44	84.29	97.49	73.05	63.84	71.82	81.10	96.50	53.88	86.39
Inf-Net	81.62	76.50	**98.32**	74.44	86.81	80.28	77.59	98.72	71.76	69.46	64.62	69.46	99.02	63.38	79.68	74.32	62.93	93.45	56.39	71.77
MobileNet	80.07	81.19	95.92	63.99	98.03	81.32	85.53	**99.62**	74.18	128.95	80.52	82.66	97.95	72.05	70.70	74.84	80.08	97.67	59.91	92.88
MobileNetv2	79.73	82.83	97.32	72.53	88.40	80.09	81.77	99.45	72.16	85.15	80.99	83.16	98.20	71.50	68.54	74.32	80.41	96.96	59.43	93.11
ShuffleNet	77.50	74.57	97.64	69.02	86.97	80.87	83.62	99.28	72.66	105.56	81.97	82.34	98.03	73.33	56.68	74.51	77.73	96.38	58.64	78.16
ShuffleNetv2	78.58	81.21	97.30	71.37	84.72	79.54	82.44	98.75	70.29	102.75	81.31	81.86	98.29	71.67	70.06	74.56	76.89	96.58	58.67	78.55
EfficientNet	78.22	80.25	97.04	70.45	75.26	73.13	73.50	99.25	60.20	133.45	81.58	80.10	98.06	72.12	64.30	73.30	80.66	97.07	58.04	96.30
ENet	79.49	81.26	97.53	71.57	96.08	79.27	79.62	99.07	70.43	101.92	77.57	76.35	97.16	68.23	67.40	74.49	74.86	96.38	57.20	85.32
ESPNet	77.45	84.18	96.48	69.30	97.04	76.79	71.30	98.67	67.68	93.58	80.32	80.53	97.52	69.36	91.84	74.75	72.06	96.96	57.77	94.58
CGNet	79.34	81.55	96.34	71.42	90.37	75.10	70.27	92.57	60.37	134.43	82.24	80.73	97.35	72.35	53.63	74.12	74.83	96.16	56.45	74.34
ESPNetv2	78.66	77.84	96.53	70.46	87.77	78.22	72.42	97.23	67.12	88.58	80.78	79.03	97.41	70.13	73.67	74.10	76.60	97.67	58.37	96.73
EDANet	78.74	82.86	96.98	70.67	88.14	80.11	79.40	98.77	72.89	70.40	79.56	76.79	97.42	68.71	70.72	73.21	73.73	96.71	55.11	84.56
LEDNet	77.41	81.69	96.93	68.74	92.49	78.46	80.96	98.47	70.41	120.74	80.34	78.74	97.90	70.10	65.77	73.46	72.27	95.14	55.09	94.19
MiniSeg	**82.15**	**84.95**	97.72	**75.91**	**74.42**	**85.31**	**90.60**	99.15	**80.06**	**58.46**	**84.49**	**85.06**	**99.05**	**76.27**	**51.06**	**78.33**	79.62	**97.71**	**64.84**	**71.69**

Table 5: Comparison between MiniSeg and previous state-of-the-art segmentation methods.

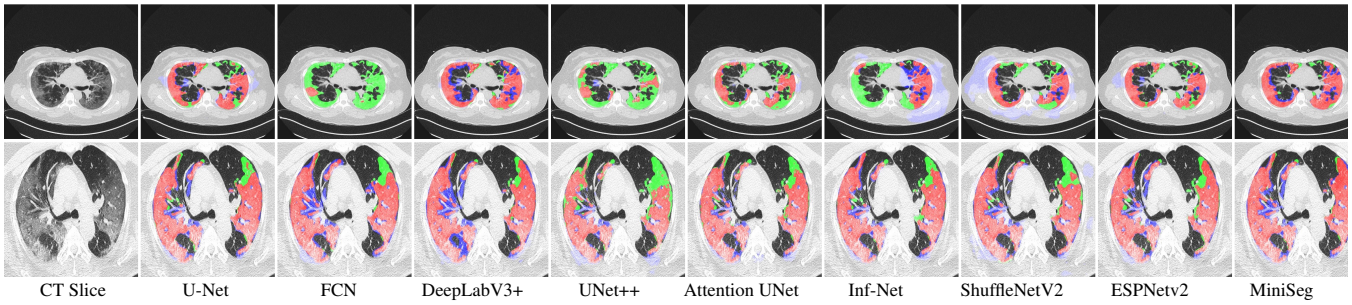

Figure 4: Visual comparison between MiniSeg and other methods. Red: true positive; Green: false negative; Blue: false positive.

ment that is of high importance in the current severe situation of COVID-19.

Qualitative Comparison. To explicitly show the superiority of MiniSeg, we provide a qualitative comparison between MiniSeg and eight state-of-the-art methods in Fig. 4. We select some representative images from the above datasets. This visual comparison further indicates that MiniSeg outperforms baseline methods remarkably.

Statistical Analysis. To further study the characteristics of MiniSeg, we perform statistical analysis on the largest COVID-19-P20 dataset. Fig. 3a and Fig. 3b illustrate the relationship between the DSC score and the infected area, or the lesion count in a CT slice, respectively. We find that MiniSeg achieves a DSC score larger than 0.7 for most CT slices regardless of the infected area. The medium DSC is above 0.8 regardless of the lesion counts. This suggests that MiniSeg is robust to different cases for COVID-19 infected area segmentation.

Conclusion

In this paper, we focus on segmenting COVID-19 infected areas from chest CT slices. To address the lack of COVID-19 training data and meet the efficiency requirement for the deployment of computer-aided COVID-19 screening systems, we propose an extremely minimum network, *i.e.,* MiniSeg, for accurate and efficient COVID-19 infected area segmentation. MiniSeg adopts a novel multi-scale learning module, *i.e.,* the **A**ttentive **H**ierarchical **S**patial **P**yramid (**AHSP**) module, to ensure its accuracy under the constraint of the extremely minimum network size. To extensively compare MiniSeg with previous state-of-the-art image segmentation methods and promote future research on COVID-19 infected area segmentation, we build a comprehensive benchmark that would be useful for future research. The comparison between MiniSeg and state-of-the-art image segmentation methods demonstrates that MiniSeg not only achieves the best performance but also has high efficiency, making MiniSeg suitable for practical deployment.

References

Ai, T.; Yang, Z.; Hou, H.; Zhan, C.; Chen, C.; Lv, W.; Tao, Q.; Sun, Z.; and Xia, L. 2020. Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases. *Radiology* 200642.

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2017. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE TPAMI* 39(12): 2481–2495.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018a. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE TPAMI* 40(4): 834–848.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* .

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018b. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 801–818.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *IEEE CVPR*, 1251–1258.

Cohen, J. P.; Morrison, P.; Dao, L.; Roth, K.; Duong, T. Q.; and Ghassemi, M. 2020. COVID-19 Image Data Collection: Prospective Predictions Are the Future. *arXiv preprint arXiv:2006.11988* .

Fan, D.-P.; Zhou, T.; Ji, G.-P.; Zhou, Y.; Chen, G.; Fu, H.; Shen, J.; and Shao, L. 2020. Inf-Net: Automatic COVID-19 Lung Infection Segmentation from CT Images. *IEEE TMI* 39(8): 2626–2637.

Fang, Y.; Zhang, H.; Xie, J.; Lin, M.; Ying, L.; Pang, P.; and Ji, W. 2020. Sensitivity of chest CT for COVID-19: Comparison to RT-PCR. *Radiology* 200432.

Fu, J.; Liu, J.; Tian, H.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *IEEE CVPR*, 3146–3154.

Gozes, O.; Frid-Adar, M.; Greenspan, H.; Browning, P. D.; Zhang, H.; Ji, W.; Bernheim, A.; and Siegel, E. 2020. Rapid AI development cycle for the coronavirus (COVID-19) pandemic: Initial results for automated detection & patient monitoring using deep learning CT image analysis. *arXiv preprint arXiv:2003.05037* .

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *IEEE ICCV*, 1026–1034.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* .

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *IEEE CVPR*, 4700–4708.

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2019. CCNet: Criss-cross attention for semantic segmentation. In *IEEE ICCV*, 603–612.

Inui, S.; Fujikawa, A.; Jitsu, M.; Kunishima, N.; Watanabe, S.; Suzuki, Y.; Umeda, S.; and Uwabe, Y. 2020. Chest CT findings in cases from the cruise ship "Diamond Princess" with coronavirus disease 2019 (COVID-19). *Radiology: Cardiothoracic Imaging* 2(2): e200110.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 448–456.

Jenssen, H. B. 2020. COVID-19 CT segmentation dataset. http://medicalsegmentation.com/covid19/. Accessed 04 10, 2020.

Jin, J.; Dundar, A.; and Culurciello, E. 2015. Flattened convolutional neural networks for feedforward acceleration. In *ICLR*.

Jun, M.; Cheng, G.; Yixin, W.; Xingle, A.; Jiantao, G.; Ziqi, Y.; Minqing, Z.; Xin, L.; Xueyuan, D.; Shucheng, C.; et al. 2020. COVID-19 CT Lung and Infection Segmentation Dataset. *Zenodo, Apr* 20.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Lee, C.-Y.; Xie, S.; Gallagher, P.; Zhang, Z.; and Tu, Z. 2015. Deeply-supervised nets. In *AISTATS*, 562–570.

Li, L.; Qin, L.; Xu, Z.; Yin, Y.; Wang, X.; Kong, B.; Bai, J.; Lu, Y.; Fang, Z.; Song, Q.; et al. 2020a. Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. *Radiology* 200905.

Li, X.; Zhao, H.; Han, L.; Tong, Y.; Tan, S.; and Yang, K. 2020b. Gated Fully Fusion for Semantic Segmentation. In *AAAI*, 11418–11425.

Liu, Y.; Cheng, M.-M.; Fan, D.-P.; Zhang, L.; Bian, J.; and Tao, D. 2018a. Semantic edge detection with diverse deep supervision. *arXiv preprint arXiv:1804.02864* .

Liu, Y.; Cheng, M.-M.; Hu, X.; Bian, J.-W.; Zhang, L.; Bai, X.; and Tang, J. 2019. Richer Convolutional Features for Edge Detection. *IEEE TPAMI* 41(8): 1939–1946.

Liu, Y.; Jiang, P.-T.; Petrosyan, V.; Li, S.-J.; Bian, J.; Zhang, L.; and Cheng, M.-M. 2018b. DEL: Deep Embedding Learning for Efficient Image Segmentation. In *IJCAI*, 864–870.

Liu, Y.; Wu, Y.-H.; Ban, Y.; Wang, H.; and Cheng, M.-M. 2020. Rethinking computer-aided tuberculosis diagnosis. In *IEEE CVPR*, 2646–2655.

Lo, S.-Y.; Hang, H.-M.; Chan, S.-W.; and Lin, J.-J. 2019. Efficient dense modules of asymmetric convolution for real-time semantic segmentation. In *ACM Multimedia Asia*, 1–6.

Ma, N.; Zhang, X.; Zheng, H.-T.; and Sun, J. 2018. ShuffleNet v2: Practical guidelines for efficient CNN architecture design. In *ECCV*, 116–131.

Mehta, S.; Rastegari, M.; Caspi, A.; Shapiro, L.; and Hajishirzi, H. 2018. ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *ECCV*, 552–568.

Mehta, S.; Rastegari, M.; Shapiro, L.; and Hajishirzi, H. 2019. ESPNetv2: A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network. In *IEEE CVPR*, 9190–9200.

Morozov, S.; Andreychenko, A.; Pavlov, N.; Vladzymyrskyy, A.; Ledikhova, N.; Gombolevskiy, V.; Blokhin, I.; Gelezhe, P.; Gonchar, A.; Chernina, V.; et al. 2020. MosMedData: Chest CT Scans with COVID-19 Related Findings. *medRxiv* .

Narin, A.; Kaya, C.; and Pamuk, Z. 2020. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849* .

Oktay, O.; Schlemper, J.; Folgoc, L. L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N. Y.; Kainz, B.; et al. 2018. Attention U-Net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999* .

Paszke, A.; Chaurasia, A.; Kim, S.; and Culurciello, E. 2016. ENet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147* .

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch. In *NIPS Workshop*, 1–4.

Pohlen, T.; Hermans, A.; Mathias, M.; and Leibe, B. 2017. Full-resolution residual networks for semantic segmentation in street scenes. In *IEEE CVPR*, 4151–4160.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 234–241.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. ImageNet large scale visual recognition challenge. *IJCV* 115(3): 211–252.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE CVPR*, 4510–4520.

Shelhamer, E.; Long, J.; and Darrell, T. 2017. Fully Convolutional Networks for Semantic Segmentation. *IEEE TPAMI* 39(4): 640–651.

Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014. Deep learning face representation by joint identification-verification. In *NIPS*, 1988–1996.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *AAAI*, 4278–4284.

Tan, M.; and Le, Q. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *ICML*, 6105–6114.

Tian, Z.; He, T.; Shen, C.; and Yan, Y. 2019. Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. In *IEEE CVPR*, 3126–3135.

Wang, L.; and Wong, A. 2020. COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest Radiography Images. *arXiv preprint arXiv:2003.09871* .

Wang, W.; Xu, Y.; Gao, R.; Lu, R.; Han, K.; Wu, G.; and Tan, W. 2020. Detection of SARS-CoV-2 in different types of clinical specimens. *J. American Medical Association* .

Wang, Y.; Zhou, Q.; Liu, J.; Xiong, J.; Gao, G.; Wu, X.; and Latecki, L. J. 2019. LEDNet: A Lightweight Encoder-Decoder Network for Real-Time Semantic Segmentation. In *IEEE ICIP*, 1860–1864. IEEE.

Wu, T.; Tang, S.; Zhang, R.; and Zhang, Y. 2018. CGNet: A light-weight context guided network for semantic segmentation. *arXiv preprint arXiv:1811.08201* .

Xu, X.; Jiang, X.; Ma, C.; Du, P.; Li, X.; Lv, S.; Yu, L.; Chen, Y.; Su, J.; Lang, G.; et al. 2020. Deep learning system to screen coronavirus disease 2019 pneumonia. *arXiv preprint arXiv:2002.09334* .

Yang, M.; Yu, K.; Zhang, C.; Li, Z.; and Yang, K. 2018. DenseASPP for semantic segmentation in street scenes. In *IEEE CVPR*, 3684–3692.

Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; and Sang, N. 2018a. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 325–341.

Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; and Sang, N. 2018b. Learning a discriminative feature network for semantic segmentation. In *IEEE CVPR*, 1857–1866.

Yuan, Y.; and Wang, J. 2018. OCNet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916* .

Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; and Agrawal, A. 2018a. Context encoding for semantic segmentation. In *IEEE CVPR*, 7151–7160.

Zhang, J.; Xie, Y.; Li, Y.; Shen, C.; and Xia, Y. 2020. COVID-19 Screening on Chest X-ray Images Using Deep Learning based Anomaly Detection. *arXiv preprint arXiv:2003.12338* .

Zhang, X.; Zhou, X.; Lin, M.; and Sun, J. 2018b. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *IEEE CVPR*, 6848–6856.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *IEEE CVPR*, 2881–2890.

Zhao, J.; Zhang, Y.; He, X.; and Xie, P. 2020. COVID-CT-Dataset: A CT scan dataset about COVID-19. *arXiv preprint arXiv:2003.13865* .

Zhou, Z.; Siddiquee, M. M. R.; Tajbakhsh, N.; and Liang, J. 2018. UNet++: A nested U-Net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 3–11. Springer.

Zhu, Z.; Xu, M.; Bai, S.; Huang, T.; and Bai, X. 2019. Asymmetric non-local neural networks for semantic segmentation. In *IEEE ICCV*, 593–602.