Published in final edited form as:

Med Image Anal. 2023 February ; 84: 102726. doi:10.1016/j.media.2022.102726.

Segmentation ability map: Interpret deep features for medical image segmentation

Sheng He[a,*,1], **Yanfang Feng**[b,1], **P. Ellen Grant**[a], **Yangming Ou**[a,*]

[a]Boston Children's Hospital and Harvard Medical School, 300 Longwood Ave., Boston, MA, USA

[b]Massachusetts General Hospital and Harvard Medical School, 55 Fruit St., Boston, MA, USA

Abstract

Deep convolutional neural networks (CNNs) have been widely used for medical image segmentation. In most studies, only the output layer is exploited to compute the final segmentation results and the hidden representations of the deep learned features have not been well understood. In this paper, we propose a prototype segmentation (ProtoSeg) method to compute a binary segmentation map based on deep features. We measure the segmentation abilities of the features by computing the Dice between the feature segmentation map and ground-truth, named as the segmentation ability score (SA score for short). The corresponding SA score can quantify the segmentation abilities of deep features in different layers and units to understand the deep neural networks for segmentation. In addition, our method can provide a mean SA score which can give a performance estimation of the output on the test images without ground-truth. Finally, we use the proposed ProtoSeg method to compute the segmentation map directly on input images to further understand the segmentation ability of each input image. Results are presented on segmenting tumors in brain MRI, lesions in skin images, COVID-related abnormality in CT images, prostate segmentation in abdominal MRI, and pancreatic mass segmentation in CT images. Our method can provide new insights for interpreting and explainable AI systems for medical image segmentation. Our code is available on: https://github.com/shengfly/ProtoSeg.

Keywords

Medical image segmentation; Prototype segmentation; U-net; Interpreting and explainable AI

1. Introduction

Medical image segmentation aims to train a machine learning model (such as the deep neural network Ronneberger et al., 2015) to learn the features of target objects from expert-annotations and apply it to test images. Deep convolutional neural networks are popular for medical image segmentation (Milletari et al., 2016; Zhou et al., 2019; Wang et al., 2020; van

[*]Corresponding authors. heshengxgd@gmail.com (S. He), yangming.ou@childrens.harvard.edu (Y. Ou).

[1]These authors contributed equally to this work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Rijthoven et al., 2020) and a typical neural network contains an input layer which receives the medical images as inputs, several convolutional layers which extract deep features and an output layer which provides the final segmentation results. Deep features are the activation maps from the hidden convolutional layers, indicating the locations and feature strength of hidden units (or filters) in deep neural networks (Mou et al., 2020). The neural network can decompose the input image into different deep features over layers toward the final output layer to separate the object and background pixels/voxels (such as the negative class or the non-annotated tissue on medical images) in the input image. The segmentation ability measures the ability of a feature map to separate the object and background at each pixel/voxel. The intensity values of the object and background pixels/voxels on the input images are usually not separable which has a low segmentation ability. However, the intensity values of the segmentation map from the last layer of the neural network with Sigmoid/Softmax operation are separable which has a high segmentation ability. Thus, there is a transition of the segmentation ability from the input image to the output of the neural network. One question is raised: **where (or from which layer) are the features of the object and background regions separable or how the segmentation ability transits from the input image to the output of the neural network?**

Answering this question requires us to quantify and visualize the object regions on deep features in hidden layers of the neural network. Quantitatively measuring the segmentation ability of the hidden deep features is useful for understanding or interpreting the neural network for segmentation. However, only deep features of the output layer can be quantitatively interpreted because it represents the segmentation result which can be evaluated based on the ground-truth. The deep features from the hidden convolutional layers of the segmentation neural network are not well understood. The aim of this paper is to develop a tool (1) for data scientists or model developers to understand the neural networks and provide some insights for improving or developing the segmentation neural networks and (2) for end users to understand how U-Net makes the segmentation by different layers to enable human users to understand and appropriately trust the transparent machine learning model (Arrieta et al., 2020).

One indirect method for understanding the hidden deep features is to use the attention mechanism to highlight salient deep features of the target task (Schlemper et al., 2019; Liu et al., 2020) by visualizing the learned attention (Gu et al., 2020). However, attention cannot be used for quantifying the importance of each deep feature or understanding how the decision is obtained on each pixel/voxel. In addition, it can only highlight the salient regions on these layers where attention is used.

To solve this problem, we propose a method to exploit the rich information contained in intermediate deep features on neural networks for segmentation. Ideally, the distance among features from the pixels/voxels in the same class (object/background) should be small while the distance among features from the pixel/voxels in different classes should be high. Based on this assumption, we propose a simple *prototype segmentation* (ProtoSeg for short) method to compute a binary segmentation map on deep features and then measure the segmentation ability by comparing it to the ground-truth. Computing a binary segmentation is very useful for understanding the deep features in neural networks (Zhou et al., 2018).

Given a deep feature, the prototypes of the object and background regions are computed as the mean of the feature values guided by the initial segmentation from the output of the neural network. After that, all pixels/voxels on the feature map can be segmented based on the prototypes, yielding a feature segmentation map, named **segmentation ability map** (or SAM for short). The segmentation ability can be measured by any metric used for segmentation evaluation between the SAM and the ground-truth. In this paper, we use the well-established Dice score to measure the segmentation ability of SAM, named segmentation ability score (SA score). A high SA score means a high segmentation ability which indicates the SAM of the deep feature close to the ground-truth.

The proposed ProtoSeg method is a plug-and-play module and is efficient without parameters. It has three advantages. First, it aims to understand the segmentation ability of deep features in different layers of the neural network. A powerful deep feature should have a high segmentation ability and output a segmentation map that is close to the ground-truth, indicating that features on object regions are different from features on normal ones. Second, the proposed ProtoSeg is differentiable, which can be used in different ways to measure the segmentation ability of different deep features: offline ProtoSeg and online ProtoSeg. For offline ProtoSeg, it can compute the SAM on any deep feature extracted on the trained neural network for interpretation. For online ProtoSeg, the segmentation ability computed by the ProtoSeg is differentiable and can be jointly trained to increase the segmentation ability of hidden features. Third, the SAM can be used to estimate the confidence scores of the network's output from different input images by computing the mean SA score of the neurons on the last two layers.

The main contributions of this paper are summarized as follows:

- We propose a plug-and-play ProtoSeg method to compute the SAM on any deep feature map. The ProtoSeg method is parameter-free, simple, differentiable and computationally efficient.
- A SA score can be computed based on the binarized SAM map to interpret the deep features of the neural network.
- The proposed ProtoSeg can be used in different ways: offline and online. For online ProtoSeg, the SA score can be used in the training loss to improve the segmentation ability without decreasing the final accuracy.
- We apply the proposed method to understand the well-established U-Net on five different datasets, providing some insights for understanding the U-Net for pixel/voxel-wise segmentation.

2. Background and motivation

Interpreting deep networks is important for explainable machine learning and medical image segmentation. Attention is usually used for understanding deep features which consists of the channel attention (Hu et al., 2018) for highlighting the channels in each layer, spatial attention (Schlemper et al., 2019) for visualizing the salient locations and the hybrid of spatial and channel attention (Lei et al., 2020). A comprehensive attention-based neural network is proposed in Gu et al. (2020) for explainable medical image segmentation, including spatial, channel and scale attentions. The limitation of using attention for interpretation is that it can only visualize the salient regions computed by the attention instead of quantifying the segmentation ability of the deep features. Features with high attention may not necessarily be most useful for separating objects from background regions. In addition, the salient regions can only be computed on the layers where attention is applied. Thus, the limitation is that we cannot quantitatively compare attentions among different layers in the deep neural networks to understand the segmentation ability of the whole neural network.

For image recognition, deep features can be interpreted by the network dissection (Zhou et al., 2018) or concept whitening (Chen et al., 2020) which aim to understand the relationship between the activation maps (deep features) of every internal convolutional unit and the concepts defined by humans. Another method of interpreting the deep features is the class activation map (CAM) which can be computed directly on the last convolutional layer (Zhou et al., 2016) or any layers using gradients (Selvaraju et al., 2017).

However, it is not straightforward to apply these methods for interpreting the deep features learned for image segmentation. The reasons are two-fold.

- First, *image recognition is a image-to-class problem while image segmentation is a image-to-image problem.* For image recognition, a neural network makes decisions based on the whole input image. The size of the activation map for each class is the same as the size of the input image because almost all pixels/voxels are involved in the class recognition to some extent. Thus, the binarized feature map (Zhou et al., 2018) is to highlight some semantic regions which are meaningful for the corresponding recognized class. However, image segmentation is a pixel/voxel-level classification problem which requires the neural network to recognize the label on each pixel/voxel. Therefore, the size of the activation map for each pixel/voxel is squeezed to that pixel/voxel. From this point of view, the class activation map for each pixel/voxel is also a pixel/voxel, instead of a feature map.
- Second, *all pixels/voxels of the input image are assumed to have one label for image recognition while they have different labels (object/background) for image segmentation.* For image classification, all pixels are assumed to have the same label thus the values computed by the CAM (Zhou et al., 2016) or Grad-CAM (Selvaraju et al., 2017) in the feature map indicate which pixels are more important for the final decision. In other words, the values within one feature map are comparable since they work together to make one single decision. However, for image segmentation, the network makes the decision based on each pixel and the values of the feature map do not reflect the importance of each pixel since different pixels carry different labels. Thus, the separation ability of the deep feature is more important than the absolute values. Overall, the absolute values of the deep features are meaningful for image recognition while the separation ability of the deep features is meaningful for image segmentation.

Fig. 1(b)–(d) give examples of three deep feature maps extracted on different units of convolutional layers on the trained neural networks from the input images (as shown in Fig. 1(a) with the ground-truth object masks shown in red). We show that features with the high activated value on the background pixels/voxels (Fig. 1(b)), on the object pixels/voxels (Fig. 1(c)) and on both the object and background pixel/voxels (Fig. 1(d)). The activated values (as measured by the CAM Zhou et al., 2016) for objects are only high in Fig. 1(c). However, the aim of segmentation is to separate the object and background pixels/voxels in the input images. Thus, the feature maps on Fig. 1(b) are also discriminative since the object regions (with small responses) can be separated from the background regions (with large responses). In fact, a negative weight can be applied on feature maps on Fig. 1(b) to highlight the object regions in the subsequent layers. Therefore, the activated values in feature maps do not necessarily translate into the ability for segmentation but the segmentation ability of deep features is important for the final segmentation.

Another observation from Fig. 1 is that the segmentation abilities of the deep features from different units are different. For example, the segmentation abilities of the feature map shown on Fig. 1(b) and (c) are greater than the segmentation abilities of the feature map shown on Fig. 1(d). In general, the segmentation abilities of deep features on different layers or on different units from the same layer are different and quantifying the segmentation abilities is important to understand the deep features and further understand the whole neural network for medical image segmentation.

3. ProtoSeg: Prototype segmentation on deep features

In this section, we describe the proposed plug-and-play and parameter-free prototype segmentation (ProtoSeg) method which can compute a binary segmentation map on a given deep feature extracted from the neural network.

A deep neural network can be divided into two parts: a feature extractor $\mathbf{E_f}$ with parameters θ_f and a segmentation predictor $\mathbf{E_s}$ with parameters θ_S. Given the input image x, a deep feature map f can be computed by: $f = \mathbf{E_f}(x, \theta_f)$ and the binarized segmentation output $\mathscr{B}$ can be obtained by $\mathscr{B} = \mathbf{E}_s(f, \theta_s)$. Our aim is to quantify the segmentation ability of the deep feature f by learning the $\mathbf{E_f}$ and $\mathbf{E_s}$ simultaneously, such that the segmentation predictor $\mathbf{E_s}$ can predict the label accurately.

3.1. Parameter-free prototype segmentation

We denote a learned deep feature tensor f with the size of $H \times W \times C$ (H, W and C are the height, width and number of channels, respectively). As we mentioned above, a discriminative feature representation should have a small distance among pixels/voxels with the same label and should have a large distance among pixels/voxels with different labels. This can be described as:

$$\underset{\forall f_i, f_j, L(f_i) = L(f_j)}{\mathcal{D}(f_i, f_j)} < \underset{\forall f_i, f_j, L(f_i) \neq L(f_j)}{\mathcal{D}(f_i, f_j)} \tag{1}$$

where f_i and f_j are the features at pixels/voxels i and j respectively. $L(f_i)$ and $L(f_j)$ are the labels of the feature f_i and f_j. $\mathscr{D}$ is the distance function, such as the Euclidean distance used in this paper. Ideally, the distance of any two pixels/voxels can be computed. However, directly computing the distance from any given features f_i and f_j is inefficient since the size of feature maps is usually high which requires a large memory and a long computing time.

In this paper, we resort to the prototype learning method (Snell et al., 2017) to compute the distance among different locations in the deep features. If the prototype (the center of the feature) for object and background regions are obtained, the Eq. (1) can be written as:

$$\underset{\forall f_i, \mathbf{c}_k, L(f_i) = L(\mathbf{c}_k)}{\mathscr{D}(f_i, \mathbf{c}_k)} < \underset{\forall f_i, \mathbf{c}_k, L(f_i) \neq L(\mathbf{c}_k)}{\mathscr{D}(f_i, \mathbf{c}_k)} \tag{2}$$

where $\mathbf{c}_k, k \in \{t, b\}$ are the prototypes of the object ($k = t$) or background regions which contain normal pixels/voxels ($k = b$). Eq. (2) shows that the distance of the features should be close to the center with the same label while far away from the center with different labels. For example, if f_i is from the object region, we have:

$$\mathscr{D}(f_i, \mathbf{c}_t) < \mathscr{D}(f_i, \mathbf{c}_b) \tag{3}$$

which can be normalized by dividing the sum of these two distances:

$$\frac{\mathscr{D}(f_i, \mathbf{c}_t)}{\mathscr{D}(f_i, \mathbf{c}_t) + \mathscr{D}(f_i, \mathbf{c}_b)} < \frac{\mathscr{D}(f_i, \mathbf{c}_b)}{\mathscr{D}(f_i, \mathbf{c}_t) + \mathscr{D}(f_i, \mathbf{c}_b)} \tag{4}$$

When using the softmax function on both sides, we can obtain:

$$\frac{\exp(-\mathscr{D}(f_i, \mathbf{c}_t))}{\exp(-(\mathscr{D}(f_i, \mathbf{c}_t) + \mathscr{D}(f_i, \mathbf{c}_b)))} > \frac{\exp(-\mathscr{D}(f_i, \mathbf{c}_b))}{\exp(-(\mathscr{D}(f_i, \mathbf{c}_t) + D(f_i, \mathbf{c}_b)))} \tag{5}$$

The Eq. (5) can be described as $p(f_i, \mathbf{c}_t) > p(f_i, \mathbf{c}_b)$ for object feature f_i. Similarly, we can obtain $p(f_i, \mathbf{c}_t) < p(f_i, \mathbf{c}_b)$ for background feature f_i.

The prototype of the object $\mathbf{c}_t$ and background regions $\mathbf{c}_b$ can be computed by any clustering method, such as k-means. However, using cluster methods is not efficient since the number of pixels/voxels is usually large. In this paper, we use the output of the neural network as the initial mask $B \in [0, 1]$ to compute the prototype of the object and background regions by:

$$\begin{aligned} \mathbf{c}_t &= \sum (B_i * f_i) / \sum B_i \\ \mathbf{c}_b &= \sum ((1 - B_i) * f_i) / \sum (1 - B_i) \end{aligned} \tag{6}$$

where $B_i \in \mathscr{B}$ and $B_i = 1$ denotes as the object region while $B_i = 0$ denotes the background region. It is straightforward to generalize the proposed method into multi-label segmentation by adding new prototypes.

Once the prototype of the target and background is given, each pixel/voxel on the *i*th position can be classified based on the nearest neighbor over distance to the prototypes (Snell et al., 2017). If $p(f_i, \mathbf{c}_t) > p(f_i, \mathbf{c}_b)$, the pixel/voxel belongs to the object. Otherwise, it is from the background region. Finally, a segmentation map S_f can be obtained based on the feature map f by:

$$s_i = \begin{cases} 1, & \text{if } p_i(\mathbf{c}_t) > p_i(\mathbf{c}_b) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $s_i \in S_f$ is the segmentation on pixel/voxel *i*.

The advantage of the proposed prototype segmentation is that it is a parameter-free method and is differentiable and efficient to compute.

It aims to separate object and background regions in the feature maps on each input image, without considering their absolute value of responses. As shown in Algorithm 1, our proposed ProtoSeg method is a plug-and-play module, which can be used in any deep feature or input image.

3.2. Network interpreting

For any given deep feature f, the segmentation map S_f can be obtained by the proposed ProtoSeg method (Algorithm 1). The segmentation ability of the deep feature f can be measured by any metric used for segmentation evaluation. In this paper, we use the well-established Dice score (Milletari et al., 2016) to compute the SA score for measuring the segmentation ability:

$$\text{SA Score}(S_f, G) = \frac{2|S_f \cap G|}{|S_f| + |G|} \tag{8}$$

where S_f is the SAM obtained from the deep feature f and G is the ground-truth. A high SA score indicates that the deep feature f is discriminative and its corresponding SAM S_f is close to the ground-truth. Therefore, SA Score(S_f,G) can be used to evaluate the segmentation ability of the feature map in a CNN, which is an objective score for interpretability that is comparable across layers and networks (Zhou et al., 2018). Fig. 2 shows several examples of feature maps f and their corresponding SAM S_f computed by the proposed method. The features can be sorted by the computed SA score.

Algorithm 1

ProtoSeg Pseudocode, PyTorch-like: https://github.com/shengfly/ProtoSeg

```
# f : deep feature map
# b: initial mask in [0,1]. (0: background, 1: object)
def ProtoSeg(f,b) : # compute the segmentation map
    c1 = Prototype(f,b) # prototype of object
    c2 = Prototype(f,1−b) # prototype of background
    p1= −torch.pow(f−cl,2).sum()
    p2 = −torch.pow(f−c2,2).sum()
    m = torch.softmax([p2,p1],1) # the probability map
    #SAM = torch.argmax(m, 1) # to compute the SAM
    return m
def Prototype(f,p): # compute the prototype
    center = torch.sum(f*p)/torch.sum(p)
    return center
```

Fig. 3 shows the whole framework for network interpretation using the proposed plug-and-play ProtoSeg method, which can compute the segmentation ability on any given deep feature on the neural network. Our proposed ProtoSeg method can be divided into two methods: offline ProtoSeg and online ProtoSeg.

- **Offline ProtoSeg:** The ProtoSeg method is applied to the trained neural network to measure the segmentation ability of deep features.
- **Online ProtoSeg:** The ProtoSeg method is differentiable and the SA Score can be used as a loss as part of the training with the neural network. Adding the SA Score in the training loss can increase the segmentation ability of the deep feature.

Using the online ProtoSeg can produce the segmentation maps of the deep features which are similar to the output/ground-truth. Thus, the segmentation maps of the learned deep features are more interpretable based on their similarity with the ground-truth. For example, if the SA score is low, the corresponding deep feature has a low segmentation ability which is reflected by the ProtoSeg loss in the training.

4. Experiments of interpreting deep features

In this section, we conduct experiments to use the proposed method to interpret the intermediate representation (deep features) learned by the deep neural networks for medical image segmentation.

4.1. Datasets

We use five datasets to evaluate the proposed method. In some datasets, the objects to be segmented are lesions and the background is normal regions. In other datasets, the objects are normal organs of interest while the background contains other structures in the image. (1) BraTS (Menze et al., 2014; Bakas et al., 2017) is used for brain tumor segmentation. We use the subjects in BraTS18 as a training set and the new subjects in BraTS19 and BraTS20 as the testing set where subjects in the training set are excluded. In total, 285 subjects are used for training and 84 subjects are used for testing. Following the work (Zhou et al., 2019), we extract 2D slices from these patients with four modalities: FLAIR, T1, T1 with contrast (T1c) and T2 images, which are concatenated as the input. We train networks to segment whole tumors, which considers all tumor labels as the positive class (Zhou et al., 2019); (2) ISIC (Codella et al., 2018) is used for skin lesion segmentation. We use the 2017 ISIC challenge dataset, which contains 2000 dermoscopic images for training and 600 images for testing; (3) COVID is used for COVID-19 infection segmentation on CT images. We merge two public datasets: COVID-19 CT segmentation dataset[2] which contains 100 CT slices and COVID-19 CT scans (Ma et al., 2020) which contains 20 scans. We collect 2D slices and split them into training (1616 slices) and testing (328 slices) sets; (4) Prostate (Litjens et al., 2012) is used for whole prostate segmentation on transverse T2-weighted scans and apparent diffusion coefficient (ADC) map. We extract 2D slice from 32 scans and split them into training (375 slices) and testing (100 slices); (5) Pancreas (Dawant et al., 2007) is used for pancreatic parenchyma and pancreatic mass segmentation. We select 281 CT scans (Simpson et al., 2019) and randomly split them into training (6860 slices) and test (1692 slices) sets.

4.2. Network training

The Adam optimizer in the PyTorch package is used to train the U-Net neural network, with a batch size of 10. The learning rate is set to 0.0001 and reduced to half at every 20 epochs. Networks for all the datasets are trained with a total of 50 epochs with the widely used cross-entropy loss. To evaluate the interpretation difference between offline ProtoSeg and online ProtoSeg, we train the model without and with the loss of the SA Score. For training without the SA Score, the training loss is applied only on the output image: $\mathscr{L} = \mathscr{L}_g$. For training with the SA Score loss, we also apply the SA Score on the feature segmentation map: $\mathscr{L} = \mathscr{L}_g + \sum \text{SA Score}/N$ where N is the number of feature segmentation maps. For a fair comparison, we use the same parameter and training methods for neural networks on all five datasets. No additional re-sampling or post-processing is included (Isensee et al., 2018).

4.3. Interpreting deep features of U-Net

Although our method can be used in any neural network, we focus on the widely used segmentation neural network U-Net (Ronneberger et al., 2015) in this paper. As shown in Fig. 4, a typical U-Net contains 18 feature maps, denoted by f_i, $i = 1,2,\ldots, 18$. The spatial resolutions of these features are different due to the max-pooling and up-sampling operations. The numbers of channels of these feature maps are also various (shown in Fig. 4). As with U-Net (Ronneberger et al., 2015), all convolutional layers (except the last one) have the same parameters: kernel size is 3 × 3, stride is 1 and padding is 1. The kernel size of 1 × 1 is used on the last output layer for predicting the output which is used as an initial mask in the proposed prototype segmentation. Batch normalization (Ioffe and Szegedy, 2015) is used to accelerate the training and the default activation function is the widely used rectified linear unit (ReLU). As shown in Fig. 4, we divide the 18 deep layers into three groups: early layers which contain the first four convolutional layers, deep abstract layers which contain the layers from 5–14 layers and late layers which contain the last four layers. The feature maps on the early and late layers have a high spatial resolution but a small number of feature dimensions. On the other hand, feature maps on the deep abstract layers have a low spatial resolution but a large number of channels.

[2] http://medicalsegmentation.com/covid19/.

Deep feature maps are obtained after each layer of the U-Net. We use the proposed ProtoSeg method to compute the SAMs on these deep feature maps and then compute their corresponding SA scores referring to the ground-truth as the measurements of the segmentation ability.

4.4. Feature segmentation ability of different layers in U-Net

4.4.1. Our method can reveal the transition of the segmentation ability from early to late layers—Our proposed method can evaluate the feature segmentation ability on any given layer. In this section, we compute the segmentation ability of all 18 layers on U-Net. For deep features on each layer, the SAM is computed by the proposed ProtoSeg and the corresponding SA score is calculated based on the ground-truth. Fig. 5 shows the average SA scores on the testing set of SAMs from different layers. The figure shows that the SA score increases from early to late layers, except on the several layers on deep abstract layers which have a low spatial resolution, yielding a small SA score. Our results confirm that the segmentation ability of deep features learned for segmentation shows a transition from early to late layers in the U-Net.

4.4.2. Our method can visualize the SAMs of deep features—Our proposed method can also visualize the SAMs of deep features from different layers and Fig. 6 shows several examples from different datasets. On early layers, deep features contain detailed information of texture, boundary and spatial structure, and they are sensitive to the noise on the input images and these noises are gradually eliminated when layers on the neural network go deep. The SA score is low on deep abstract layers and the main reason is that the spatial resolution is low. However, the low spatial deep features after max-pooling can localize the object regions, representing the semantic context for segmentation. The boundaries on these deep abstract layers are smooth around the object regions and the detailed boundaries are recovered on the following late layers. When combining the features of the early and deep abstract layers, the segmentation maps from the late layers are close to the ground-truth.

4.4.3. Training the network with the SA score (online ProtoSeg) in the loss does not hurt the performance—Table 1 shows the SA scores of neural network's output B and the SA scores on the feature segmentation maps from the last layer S_{f18} (the 18th layer in Fig. 4). We choose the last layer because it provides the highest SA score (as shown in Fig. 5). There are two observations: (1) using the SA score in the training loss on the SAMs does not hurt the accuracy of the neural network. On the five datasets, there is no significant difference between the outputs B with/without using the SA score in the training loss on deep features. However, (2) the segmentation on feature maps S_{f18} using the SA score in the training loss provides a higher SA score than the output of the neural network, which demonstrates that the deep features contain rich information and the proposed ProtoSeg could find the deep features with the highest SA score. Figs. 5 and 6 show that even without the ProtoSeg loss, the intermediate features computed by different convolutional kernels/filters have some correlations with the final output. The reason would be that U-Net decomposes the input lesion-bearing image, and Fig. 6 shows that the abstract layers help reduce the noise and locate the main part of the lesions or target regions. Layer

17 picks high-intensity regions, because these regions had a high segmentation ability as measured by the proposed SA score, not because these regions had high intensities.

4.4.4. Training the network with the SA score (online ProtoSeg) in the loss can increase the segmentation ability of deep features—

Fig. 6 and Table 1 show that the segmentation ability of the deep features trained with the SA scores (online ProtoSeg) in the loss are higher than the ones trained without SA score (offline ProtoSeg), indicating that using the SA score in the loss can increase the segmentation ability of deep features on the intermediate layers of the U-Net. In addition, the SA scores of the segmentation $\mathcal{S}_{18}$ are higher than the segmentation of the output $\mathscr{B}$ on these five datasets.

4.4.5. Discussion—

Several insights can be found on Figs. 5, 6 and Table 1 for understanding the U-Net. There is a transition of the segmentation ability from the early layers to the late layers. When reducing the spatial resolution, the down-sample layers can gradually eliminate the noise and roughly localize the object regions. The detailed boundaries of the object shape are gradually recovered in the late layers. It is the first time to quantify and visualize the widely-suspected phenomenon that the encoder path in U-Net is to encode the input images into the global context to gradually eliminate the noise and the decoder path is to recover the object shape from the global context and the early feature maps. *In other words, downsampling is important for eliminating the noise or false positive.* In addition, the main reason why applying the ProtoSeg on the last feature map can provide better performance than the output of the neural network is that the prototypes of the object and background regions are learned directly from the feature maps which are discriminative and adaptive for each input image. However, the output B is computed by the linear weights which are learned from the whole dataset thus less adaptive to each image.

4.5. Feature segmentation ability of different units on the last layer

Although our proposed method can evaluate the segmentation ability of units at any layer, we focus on the last layer because the SA score of the last layer is usually higher than other layers (as shown in Fig. 5). There are 64 units on the last convolutional layer (layer 18 in Fig. 4) and it is interesting to know the segmentation ability of these individual units. We extract the deep feature on each unit and use the proposed ProtoSeg to compute the SAM and the SA score. Fig. 7 shows the SA scores of 64 different units on the last layer. The index of the units is meaningless since the weights are randomly initialized. Thus, we sort the units according to their SA scores. It clearly shows that units can be grouped into two parts according to the SA scores and only 20%–50% units have higher SA scores close to the performance of the output. Fig. 8 shows the segmentation maps of several selected units with the lowest, mediate and highest SA scores. Fig. 9 shows the average response (heat-map) of the segmentation maps computed on the units of the last layer. Object pixels/voxels have high values, indicating that most units can separate object regions from the background regions. There is a clear transition from the object regions to the background regions and the pixels/voxels on the boundary of object regions have higher uncertainty values (Nair et al., 2020).

Our results support the theory that the last layer of the U-Net is heavily oversized and only a part of units have higher discriminative scores (Hu et al., 2016). From Fig. 8 we can see that the segmentation maps of the weak units with low SA scores contain both the object and background regions. The SA score of the best unit is similar to the output of the neural network which linearly combines all 64 units on the last layer.

One possible reason is that the network tends to overfit on a group of units with high SA scores and their SAMs are close to the ground-truth on most images. The output of the neural network can be denoted as:

$$y = \sum_{i}^{N} w_i \cdot f_i + b \tag{9}$$

where w_i is the learned weight of the unit, b is the bias and f_i is the deep feature. The units can be divided into active and inertia groups:

$$y = \underbrace{\sum_{i}^{N_1} w_i \cdot f_i}_{\text{active}} + \underbrace{\sum_{j = N_1}^{N} w_i \cdot f_i}_{\text{inertia}} + b \tag{10}$$

During training, the active units provide similar results to the ground-truth which will dominate the gradient toward the weights of the active units. Meanwhile, the inertia units give a constant output $\sum_{j = N_1}^{N} w_j \cdot f_j \approx C$ (with low discriminative SA score) for both the object and background pixels/voxels. Eq. (10) becomes:

$$\begin{aligned} y &\approx \underbrace{\sum_{i}^{N_1} w_i \cdot f_i}_{\text{active}} + \underbrace{C}_{\text{inertia}} + b \\ &\approx \underbrace{\sum_{i}^{N_1} w_i \cdot f_i}_{\text{active}} + b_1 \end{aligned} \tag{11}$$

where b_1 is the bias: $b_1 = C + b$ and $N_1 < N$ is the number of active units. From Fig. 7 we can see that the number of active units varies by datasets. For example, on the ISIC dataset, there are more than 20 active units while on the Pancreas dataset in which images contain small objects there are only around 11 active units. Our results demonstrate that network pruning might be helpful for building efficient segmentation networks which can exclude the inertia units by network pruning method (Molchanov et al., 2016; Zhu and Gupta, 2017).

4.6. Evaluate the quality of segmentation outputs

Most studies for medical image segmentation only report the average Dice score for system evaluation. However, the performance can vary on different input images. For example, the performance is high on some images with high resolution and clear boundaries of the objects while the Dice score is low on some images with poor quality. Given an image from the testing set without the ground-truth, it is important to know the estimated accuracy of its segmentation output from the neural network. In other words, a confident score of each test image is important for end-users to make decisions with a “human-in-the-loop” workflow.

In practice, there is no ground-truth to compute the confident score of the predicted segmentation map in the test set. In this section, we define a measurement for roughly evaluating the quality of the output segmentation $\mathscr{B}(x)$, based on the SA score between the feature segmentation $S_f(x)$ maps and neural network output $\mathscr{B}(x)$ given the input image x. We compute the SA score of the feature segmentation on each unit of the last two layers, considering the output $\mathscr{B}(x)$ as the best approximation to the unknown ground-truth. The mean SA score $\mu(x) \in [0, 1]$ is defined as:

$$\mu(x) = \frac{1}{N}\sum_{i}^{N} \text{SA score}\left(S_{f_i}(x),\ \mathscr{B}(x)\right) \tag{12}$$

where N is the number of units on the last two layers involved in the computation and f_i is the deep feature on the ith unit.

Fig. 10 shows the relationship between the performance of the neural network and the mean SA score μ of the units from the last two convolutional layers. *The segmentation performance of the network correlates to the mean SA score μ on all five datasets.* For images with a high mean SA score μ, the Dice score of the neural network’s output $\mathscr{B}$ is also high. We also compare it with the Variance of Gradients (VoG) (Agarwal et al., 2022) which is used for ranking the test images. VoG is specifically designed for image classification. Fig. 10 shows that the correlations between the VoG score and segmentation accuracy (Dice) are smaller than the correlations between the mean SA score μ and segmentation accuracy for medical image segmentation on the five datasets. Fig. 12 shows the ranking list of images based on the mean SA score μ. Images with a low mean SA score μ are the difficult images which need more attention in practice. Thus, the mean SA score is a meaningful measurement for ranking the test images without ground-truth by difficulty and surfacing a tractable subset of the most difficult test images for human-in-the-loop auditing (Agarwal et al., 2022). Fig. 13 shows examples of the image with heat-maps and outputs from the neural network. The corresponding mean SA scores of segmentation maps on deep features and Dice scores of the neural network’s output are also shown on each image. From the example images we can see that images with small objects or smooth boundaries usually have a low mean SA score μ and the segmentation performance of the output is also low. Therefore, the performance of segmentation is mainly affected by the small objects or objects with the smooth boundaries which usually have a small mean SA score μ. Put simply, smaller and blurry objects are more difficult to segment.

Based on the above observation, the mean SA score μ can be used for sample rejection, which can configure the system to warn the end-users for the prediction on any images with the mean SA score lower than a threshold. The neural network usually has a low performance on images with small lesions or object regions with smooth boundaries and our proposed mean SA score μ can be served as a rough measurement for the end-users to pay more attention to these images. Similar to the work (Ghesu et al., 2020), we also use the term coverage, as an expected percentile of samples to be rejected for further attention to these segmentation results. For example, at a coverage of 100%, the neural network outputs its prediction on all test images, while at coverage of 80%, the neural network rejects to output the segmentation on 20% of images with the lowest mean SA score μ (Ghesu et al., 2020). By rejecting the images with small objects or unclear boundaries indicated by the mean SA score μ, the performance of the U-Net increases on the remaining test images, as shown in Table 2. In Fig. 11, we plot the segmentation accuracy of examples bucketed by the mean SA score μ, inspired by Agarwal et al. (2022). It shows that samples with the lowest percentiles of mean SA score μ have low segmentation accuracy and the segmentation performance increases with an increase in mean SA score μ, which are consistent across the five datasets. In addition, Fig. 11 shows that the VoG score is not a good predictor for the segmentation accuracy and segmentation performance decreases with an increase in VoG score.

4.7. Understanding the separableness of the input images

The separableness of the input images is defined as the segmentation ability of the input images by the color/intensity values directly. For example, the image in Fig. 14(a) has a lower separableness since the color values on the object regions are quite similar to the color values on the normal regions. It is hard to segment such lesions, even by human experts. However, the image on Fig. 14(c) has the largest separableness because color values on the lesion regions (the object) are quite different from the color values on the normal regions (the background).

The separableness is considered as the special segmentation ability of the input image, which can also be computed by the proposed ProtoSeg method. The segmentation map S_X can be computed by considering the colors/intensities of input image x as features. Thus, the performance of the segmentation map S_X indicates the quality or the separableness of the input image. Lesions in some images have high contrast and clear boundary, which are easy to be segmented, even using traditional methods based on color or intensity values, yielding a SAM with a high SA score.

In this section, we study the relationship of performance between the S_X (the segmentation map computed directly on input image x) and the network's output $\mathscr{B}(x)$, investigating how neural networks improve the segmentation on each individual image x. Fig. 15 shows the relationship between the segmentation map of input image S_X and the corresponding output of the neural network $\mathscr{B}(x)$. In most cases, U-Net can improve the segmentation performance and the Dice score of the output $\mathscr{B}(x)$ is higher than the SA score of S_X. We define the distance between the Dice scores of the S_X and $\mathscr{B}(x)$ on one image as: $d = \text{Dice}(\mathscr{B}, G) - \text{SA score}(S_x, G)$ where G is the ground-truth. For one dataset, we compute the average distance over all possible test images as: $m(d) = \sum d/N$. A high $m(d)$ means a large gain obtained by training a deep network for segmentation. Table 3 shows the Dice scores of the S_X, $\mathscr{B}$ and the $m(d)$ on different datasets. The most challenging dataset among the five datasets is the COVID segmentation, which has the lowest $m(d)$, indicating that the gain achieved by the trained U-Net is low. In addition, we have found that most images on BraTS and ISIC are easy samples in which the lesion regions can be easily separated only based on the input image pixels, yielding higher SA scores of S_X. Thus, our proposed method can be used to describe the characteristics of the dataset for segmentation.

4.8. Understanding the feature segmentation ability with noise inputs

The segmentation ability (SA) score is also a useful measurement to investigate how the segmentation ability of feature changes when the input changes, such as with the different levels of noise. This section presents the SA difference between the noise and clean input images. For each input image, we add the random noise on each pixel/voxel with different levels (the maximum value of the noise) from 0.1 to 1. The SA difference is computed by: $SA(i_n) - SA(i)$ where i_n is the noise version of the input image i. Fig. 16 shows the results on the five datasets. On BraTS, the noise affects the segmentation ability on the early and late layers of U-Net. On the other four datasets, the noise affects the segmentation ability on all layers and the abstract layers are less sensitive to noise compared to early and late layers. The results of this figure show that the effects of the noise are different when U-Net is trained on different datasets. Another finding is that the late layers are more sensitive to the noise of the input images which is introduced by the shortcut connection for the early layers in U-Net. Thus, the shortcut connection in U-Net needs to be further investigated by model developers.

5. Conclusions and future work

This paper presents a prototype segmentation (ProtoSeg) method which can compute the SAM on deep features extracted from neural networks. Our proposed method can be used to evaluate the segmentation ability of any deep feature. We have used it to evaluate the discriminative score of different layers of the most popular U-Net and the score of different units on the last layer of the U-Net. We have found that there is a transition of the segmentation ability from the early layers to the late layers and the down-sampling on the U-Net is important to eliminate the noise and localize the lesions. Fig. 6 shows that the **U-Net is actually a denoise model, which reduces the noise on the input image toward the final segmentation map.** We have further proposed to define the mean SA score μ between the output of the neural network and the feature segmentation maps to evaluate the quality of the input images. Most studies for segmentation only report the average Dice performance on the whole test images. However, the accuracy of some images (e.g., those with small lesions or lesions with smooth boundaries) is low. Without the ground-truth on the test images, it is hard for end-users to know which segmentation results need to be double-checked. Our experimental results show that the mean SA score is linearly correlated to the accuracy of segmentation from the neural network, and therefore, SA can serve as a quantity to automatically reject some images that may be subject to large segmentation errors even when the ground-truth is not available. When the low SA values trigger the rejection of these images, end-users need to pay more attention to the segmentation results of these images. In addition, we have used the proposed method to evaluate the separableness of each individual image based on a segmentation map of colors/intensities in the input images and further used it to describe the characteristics of the dataset for medical image segmentation.

Experimental results show that using the ProtoSeg loss in the training does not hurt the performance but increases the segmentation ability of the deep features. Since the ProtoSeg loss is not applied to the final output, it does not affect the accuracy of the output. Using the ProtoSeg loss can produce the segmentation maps of the learned deep features which are similar to the output/ground-truth. This provides a better interpretation by measuring the similarity between the segmentation maps and outputs. Thus, using the ProtoSeg loss can increase the interpretability of the deep features. On the other hand, a low SA score of the deep learned features indicates that it is hard to interpret those deep features since they are not similar to the output. In addition, Figs. 6 and 7 produced by the proposed method show that the features from different layers or on the same layer are redundant in U-Net and neural network pruning is needed in the future to build efficient models.

We have found that using the ProtoSeg loss does not improve the final accuracy, indicating that the intermediate losses on feature maps do not affect the performance of the segmentation. The possible reason is that adding the ProtoSeg loss only increases the interpretability of deep features which may not transfer to the accuracy of the output.

The limitations of the paper include: (1) we only used the Dice score as the SA score to measure the segmentation ability of the deep feature since the Dice score is a widely used metric for training and evaluation. Other metrics can be used, such as Jaccard index, pixelwise accuracy, sensitivity and specificity (Li et al., 2020), which will be our future work. (2) We only evaluated the proposed method on 2D network models. While we expect the conclusion to be general on 2D and 3D images, and for 2D and 3D network models, our future work includes testing this plug-and-play module into 3D networks. (3) we did not find any relation between the SA scores and the corresponding learned weights on the units of the last layer since our proposed ProtoSeg focuses on the separable ability of the features instead of the absolute response on each unit. Thus, a unit with a high SA score may have low response values and thus need a high weight to output the segmentation map.

The proposed ProtoSeg method aims to understand the learned deep features of the neural network, which can be considered as an auditing tool to provide a quantity value of the segmentation ability of the deep feature in different layers. It has the potential application to further quantitatively measure the sensitivity of input images with small variations and noise, dataset shift, out-of-sample effects, etc. For example, Fig. 16 shows how the segmentation ability of deep features in different layers changes with different levels of noise. Fig. 15 shows that the images on BraTS are easy to segment based on input intensities and Fig. 16 further verifies that it may not necessary to use a deep network since the abstract layers are less sensitive to the input changes. Other potential applications of the proposed method include the automatical evaluation of the feature maps on any network or block for segmentation. When model developers design new modules, the proposed method can be used to evaluate how the new module affects or boosts performance. Thus it can provide insights for interpreting the network or segmentation results. One of the future works is to extend the proposed ProtoSeg method to interpret the deep features of neural networks for classification which needs different prototype computations of the target class and the background. Our study motivates future work toward building more explainable AI systems for medical image segmentation, helping model developers to understand the transition of segmentation ability from input to output, and ranking the test images without ground-truth by the difficulty of segmentation and automatically suggesting the most challenging examples for human experts to review.

Acknowledgments

The work was supported, in part, by Charles A. King Trust Fellowship (SH), Massachusetts Life Science Center Bits to Bytes grant (YO), NIH/NICHD R03 HD104891 (YO), NIH/NICHD R03 HD107124 (YO), NIH/NINDS R21 NS121735 (YO), NIH/NCATS R21 TR004265 (YO), NIH/NINDS R61 NS126792 (PEG, YO).

Data availability

All data are public available

References

Agarwal C, D’souza D, Hooker S, 2022. Estimating example difficulty using variance of gradients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10368–10378.

Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García S, Gil-López S, Molina D, Benjamins R, et al. , 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion 58, 82–115.

Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby JS, Freymann JB, Farahani K, Davatzikos C, 2017. Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci. Data 4, 170117. [PubMed: 28872634]

Chen Z, Bei Y, Rudin C, 2020. Concept whitening for interpretable image recognition. Nat. Mach. Intell..

Codella NC, Gutman D, Celebi ME, Helba B, Marchetti MA, Dusza SW, Kalloo A, Liopyris K, Mishra N, Kittler H, et al., 2018. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In: International Symposium on Biomedical Imaging (ISBI 2018). pp. 168–172.

Dawant BM, Li R, Lennon B, Li S, 2007. Semi-automatic segmentation of the liver and its evaluation on the MICCAI 2007 grand challenge data set. In: 3D Segmentation in the Clinic: A Grand Challenge. Citeseer, pp. 215–221.

Ghesu FC, Georgescu B, Mansoor A, Yoo Y, Gibson E, Vishwanath R, Balachandran A, Balter JM, Cao Y, Singh R, et al. , 2020. Quantifying and leveraging predictive uncertainty for medical image assessment. Med. Image Anal 68, 101855. [PubMed: 33260116]

Gu R, Wang G, Song T, Huang R, Aertsen M, Deprest J, Ourselin S, Vercauteren T, Zhang S, 2020. CA-Net: Comprehensive attention convolutional neural networks for explainable medical image segmentation. IEEE Trans. Med. Imaging.

Hu H, Peng R, Tai Y-W, Tang C-K, 2016. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. arXiv preprint arXiv: 1607.03250.

Hu J, Shen L, Sun G, 2018. Squeeze-and-excitation networks. In: Computer Vision and Pattern Recognition, pp. 7132–7141.

Ioffe S, Szegedy C, 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456.

Isensee F, Petersen J, Klein A, Zimmerer D, Jaeger PF, Kohl S, Wasserthal J, Koehler G, Norajitra T, Wirkert S, et al., 2018. Nnu-net: Self-adapting framework for u-net-based medical image segmentation. arXiv preprint arXiv:1809. 10486.

Lei B, Huang S, Li H, Li R, Bian C, Chou Y-H, Qin J, Zhou P, Gong X, Cheng J-Z, 2020. Self-co-attention neural network for anatomy segmentation in whole breast ultrasound. Med. Image Anal. 101753.

Li X, Yu L, Chen H, Fu C-W, Xing L, Heng P-A, 2020. Transformation-consistent self-ensembling model for semisupervised medical image segmentation. IEEE Trans. Neural Netw. Learn. Syst 32 (2), 523–534.

Litjens G, Debats O, van de Ven W, Karssemeijer N, Huisman H, 2012. A pattern recognition approach to zonal segmentation of the prostate on MRI. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 413–420.

Liu L, Kurgan L, Wu F-X, Wang J, 2020. Attention convolutional neural network for accurate segmentation and quantification of lesions in ischemic stroke disease. Med. Image Anal. 65, 101791. [PubMed: 32712525]

Ma J, Wang Y, An X, Ge C, Yu Z, Chen J, Zhu Q, Dong G, He J, He Z, et al. , 2020. Towards efficient COVID-19 CT annotation: A benchmark for lung and infection segmentation. arXiv preprint arXiv:2004.12537.

Menze BH, Jakab A, Bauer S, Kalpathy-Cramer J, Farahani K, Kirby J, Burren Y, Porz N, Slotboom J, Wiest R, et al. , 2014. The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans. Med. Imaging 34 (10), 1993–2024. [PubMed: 25494501]

Milletari F, Navab N, Ahmadi S-A, 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: Fourth International Conference on 3D Vision (3DV). pp. 565–571.

Molchanov P, Tyree S, Karras T, Aila T, Kautz J, 2016. Pruning convolutional neural networks for resource efficient inference. arXiv preprint arXiv:1611.06440.

Mou L, Zhao Y, Fu H, Liu Y, Cheng J, Zheng Y, Su P, Yang J, Chen L, Frangi AF, et al. , 2020. CS2-Net: Deep learning segmentation of curvilinear structures in medical imaging. Med. Image Anal 67, 101874. [PubMed: 33166771]

Nair T, Precup D, Arnold DL, Arbel T, 2020. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. Med. Image Anal 59, 101557. [PubMed: 31677438]

van Rijthoven M, Balkenhol M, Silina K, van der Laak J, Ciompi F, 2020. HookNet: multi-resolution convolutional neural networks for semantic segmentation in histopathology whole-slide images. Med. Image Anal 101890.

Ronneberger O, Fischer P, Brox T, 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241.

Schlemper J, Oktay O, Schaap M, Heinrich M, Kainz B, Glocker B, Rueckert D, 2019. Attention gated networks: Learning to leverage salient regions in medical images. Med. Image Anal 53, 197–207. [PubMed: 30802813]

Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D, 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: International Conference on Computer Vision, pp. 618–626.

Simpson AL, Antonelli M, Bakas S, Bilello M, Farahani K, Van Ginneken B, Kopp-Schneider A, Landman BA, Litjens G, Menze B, et al., 2019. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063.

Snell J, Swersky K, Zemel R, 2017. Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, pp. 4077–4087.

Wang R, Cao S, Ma K, Zheng Y, Meng D, 2020. Pairwise learning for medical image segmentation. Med. Image Anal 67, 101876. [PubMed: 33197863]

Zhou B, Bau D, Oliva A, Torralba A, 2018. Interpreting deep visual representations via network dissection. IEEE Trans. Pattern Anal. Mach. Intell 41 (9), 2131–2145. [PubMed: 30040625]

Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A, 2016. Learning deep features for discriminative localization. In: Computer Vision and Pattern Recognition, pp. 2921–2929.

Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J, 2019. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. IEEE Trans. Med. Imaging 39 (6), 1856–1867. [PubMed: 31841402]

Zhu M, Gupta S, 2017. To prune, or not to prune: exploring the efficacy of pruning for model compression. arXiv preprint arXiv:1710.01878.

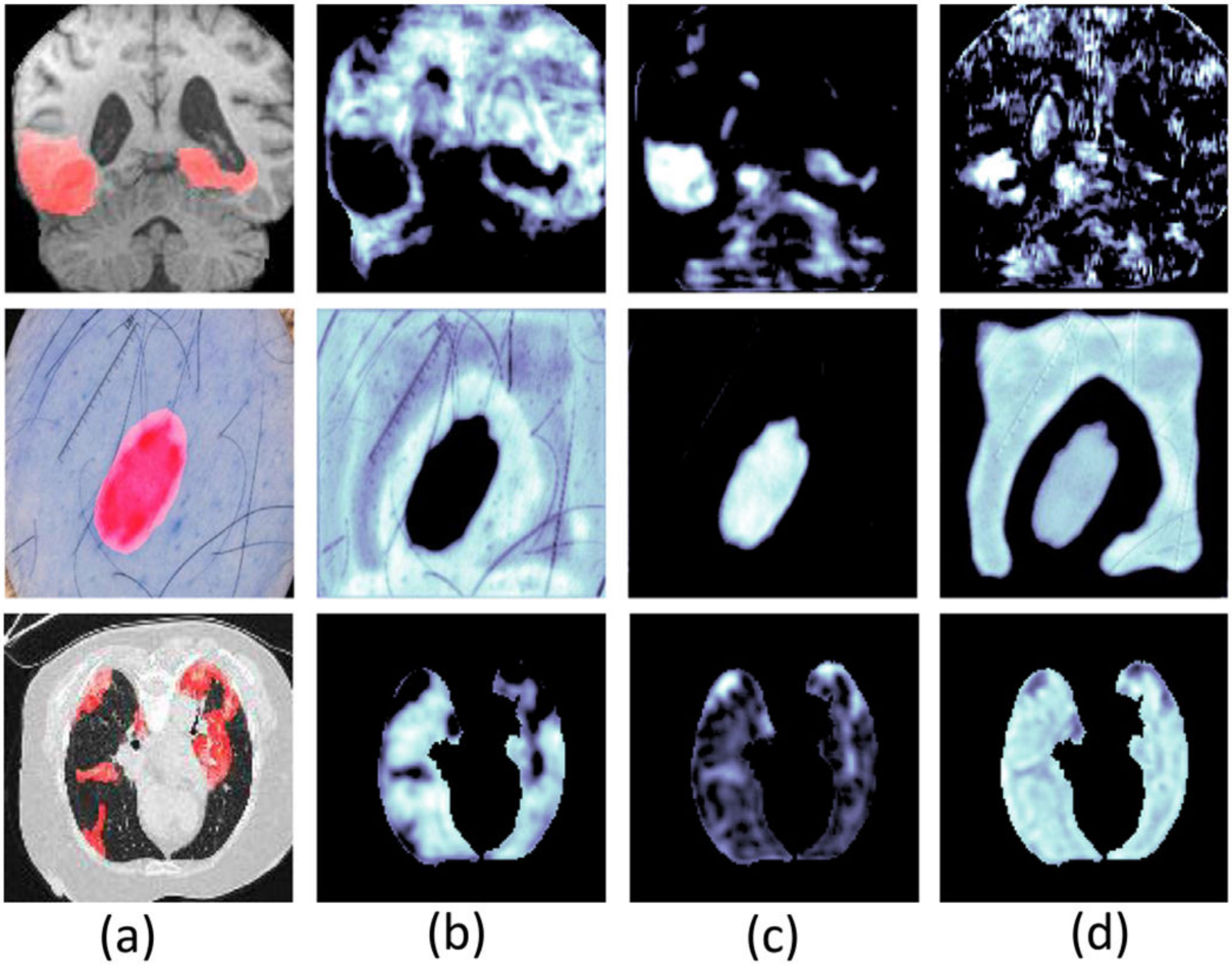


Fig. 1.

Visualization of the input images with object masks (a) and three corresponding deep features (b)–(d) extracted from three convolutional units. (b) shows that the activated values are on the background pixels/voxels, (c) shows the activated values are on the object pixels/voxels and (d) shows the activated values are on both object and background pixels/voxels.

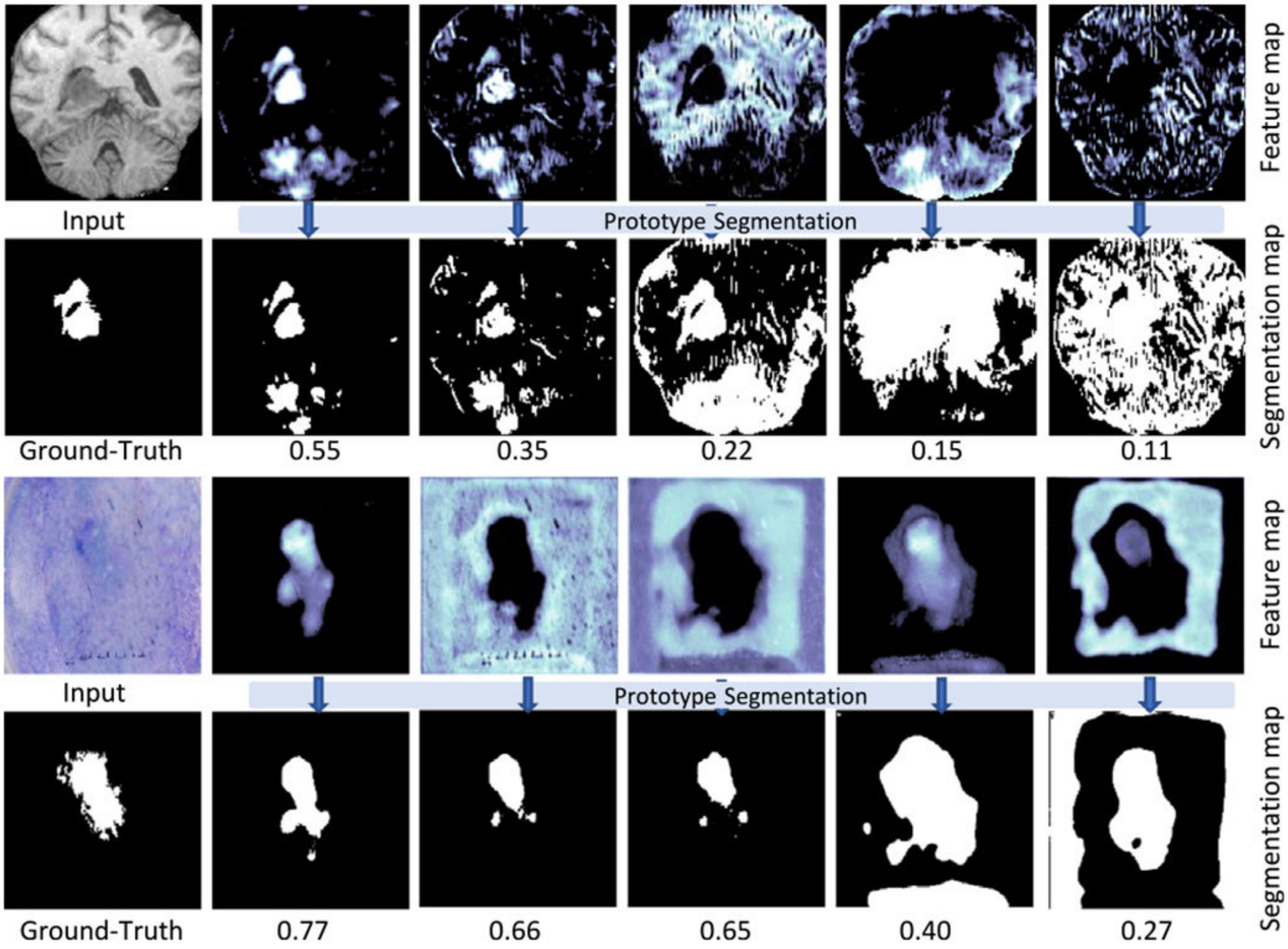


Fig. 2.

Examples of feature maps and their corresponding segmentation maps from the prototype segmentation for brain tumor (top two rows) and skin lesion (bottom two rows). The SA scores are shown on the bottom of each feature map.

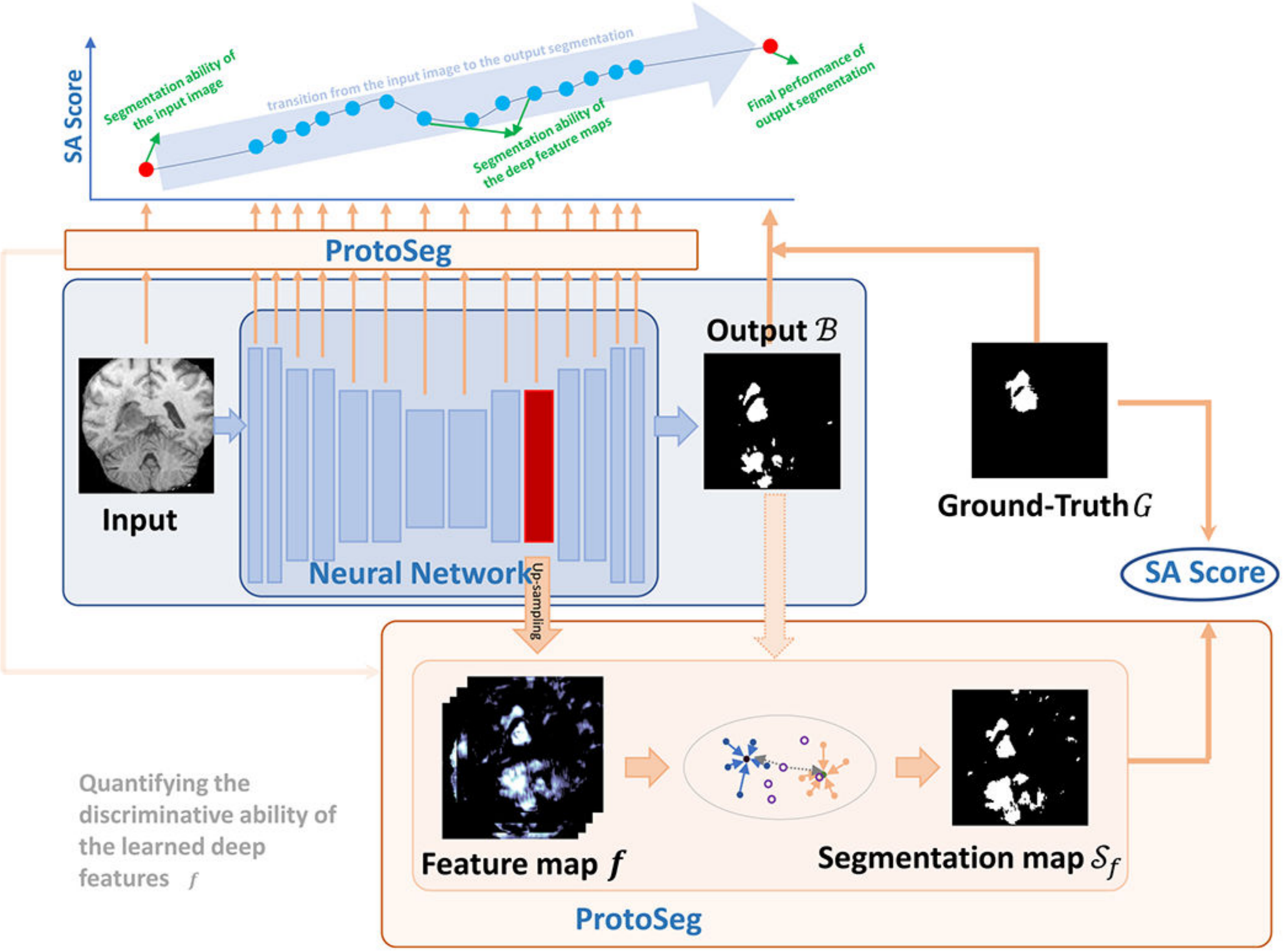

SA Score
Segmentation ability of the input image
transition from the input image to the output segmentation
Segmentation ability of the deep feature maps
Final performance of output segmentation
ProtoSeg
Output $\mathcal{B}$
Input
Neural Network
Up-sampling
Ground-Truth G
SA Score
Quantifying the discriminative ability of the learned deep features f
Feature map f
Segmentation map $\mathcal{S}_f$
ProtoSeg

Fig. 3.

The framework of the proposed method for deep feature interpretation. Given any deep feature f on the neural network, it is up-sampled to the size of the input image. Then the resized feature map is binarized by using the proposed prototype segmentation (ProtoSeg) with the initial mask B. The segmentation ability of the feature f is measured by the SA score between the segmentation map S_f and the ground-truth G. Our proposed ProtoSeg can be used in any deep feature or input image to measure their segmentation abilities and reveal the transition of the segmentation ability from the input image to the output segmentation.

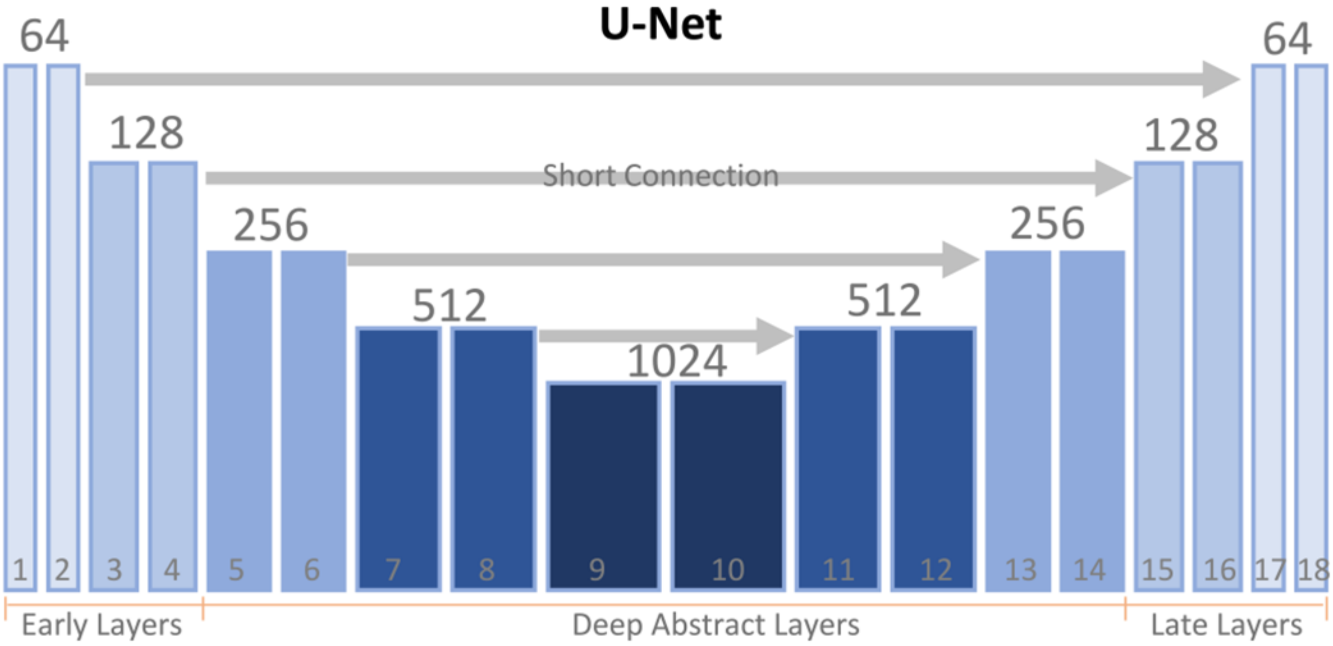


Fig. 4.

The structure of the U-Net. Typically, there are 18 convolutional layers, indexed by the number from 1 to 18. The number of channels on each layer is indicated on the top. We divide the features of each layer into three groups: early features (from 1–4 layers), deep abstract features (from 5–14 layers) and the late features (the last 4 layers).

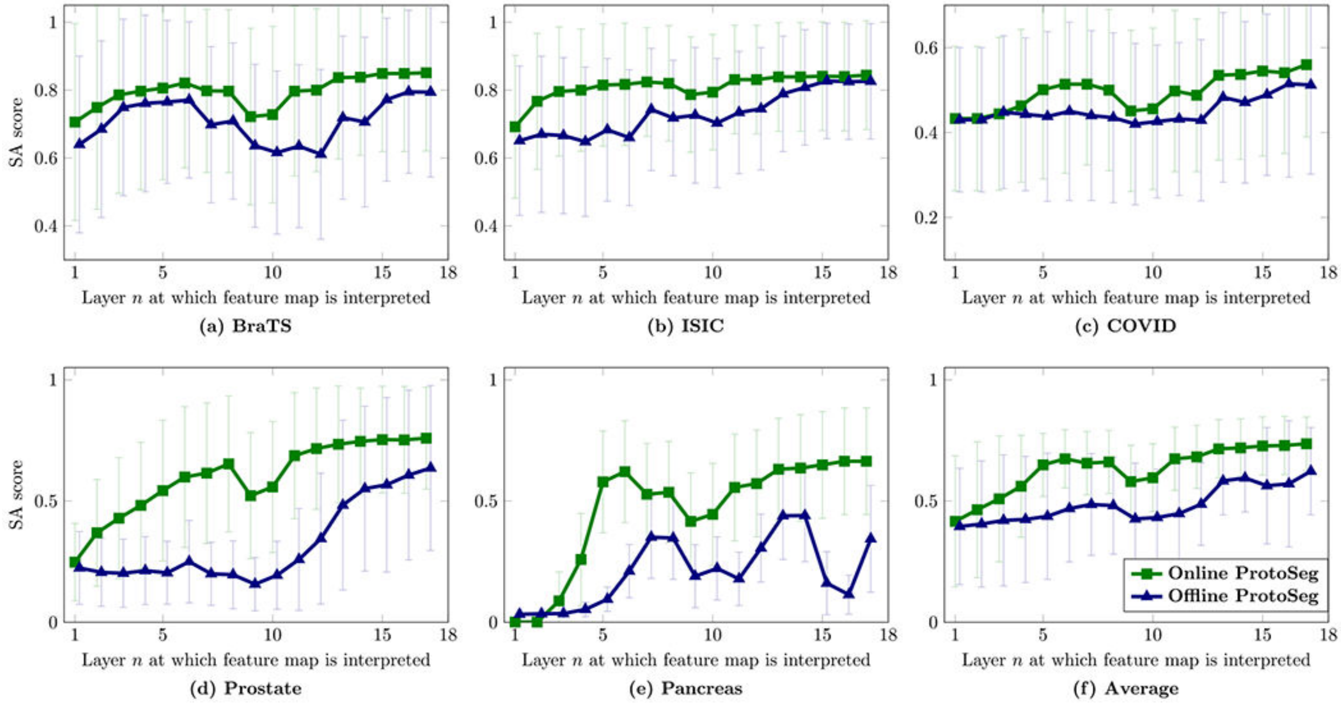


Fig. 5.

The segmentation ability (SA score) of deep features from different layers of the U-Net. Training the network with the SA score in the loss can increase the segmentation ability of deep features.

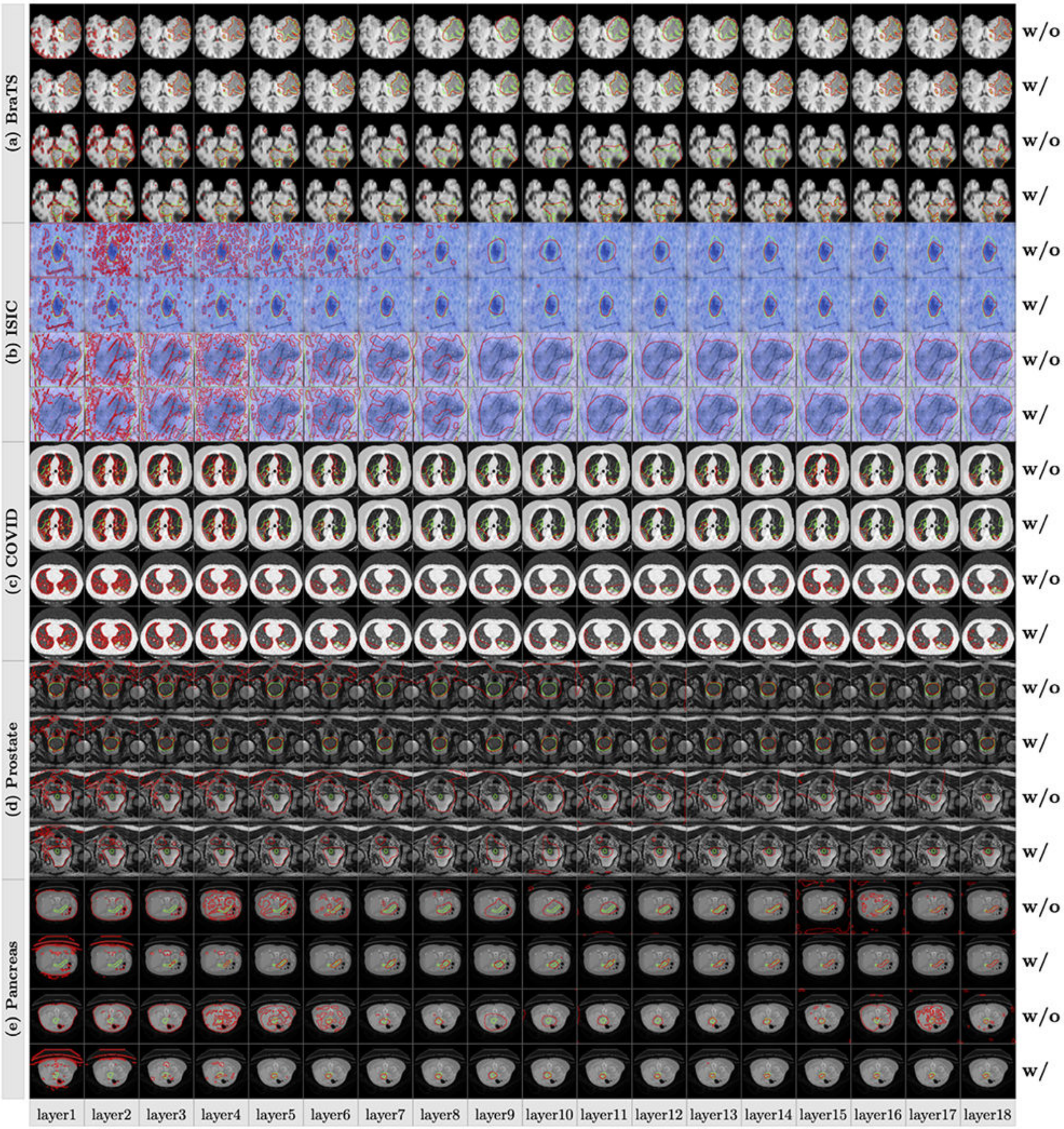


Fig. 6.

Visualizations of the segmentation maps obtained from different feature maps on layers 1–18 of the U-Net. The green and red contours denote the ground-truth and the results of the segmentation maps, w/ means the network is trained with the SA score in the training loss while w/o means without the SA score.

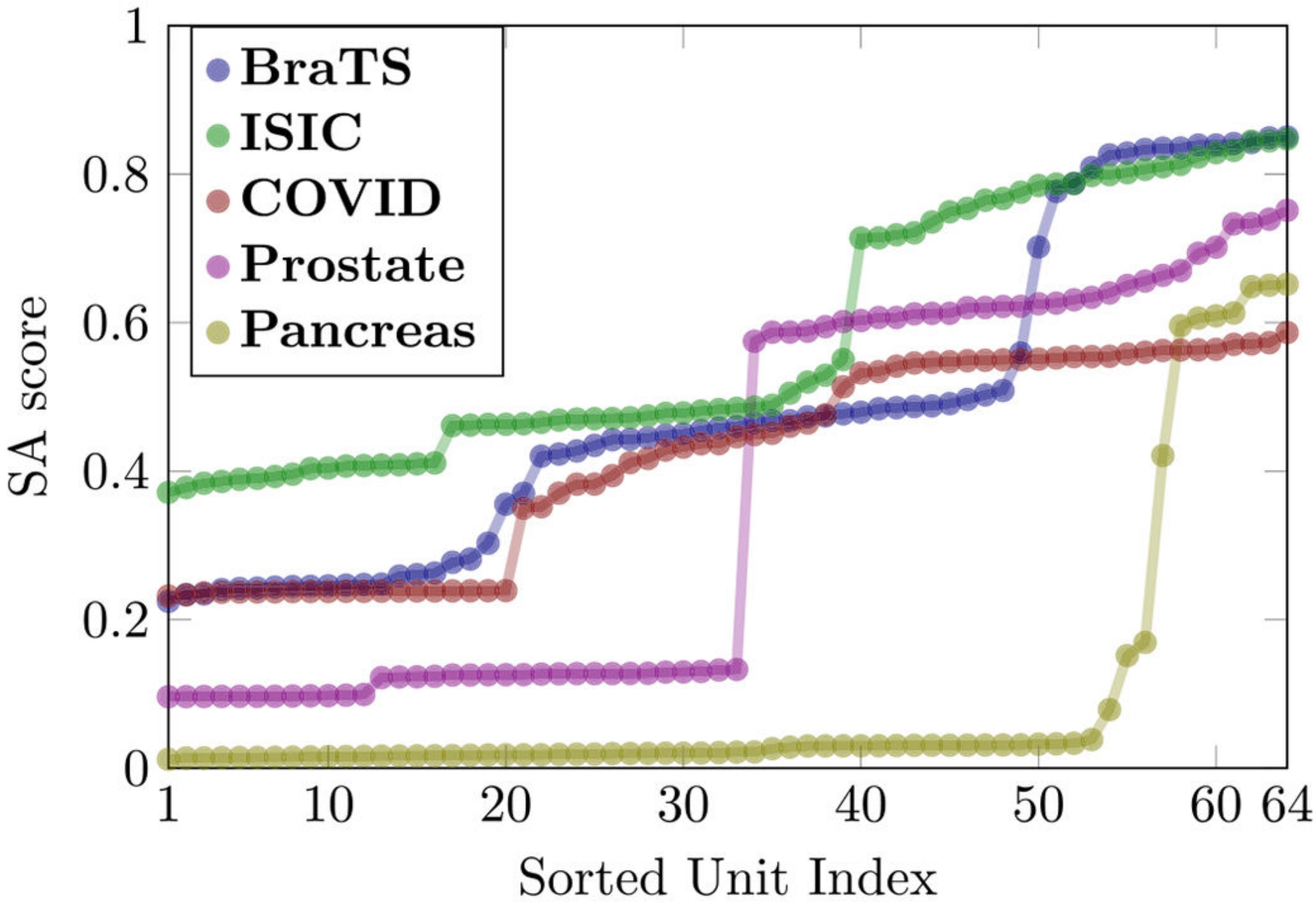


Fig. 7.

The sorted SA score of 64 units on the last convolutional layer of the U-Net.

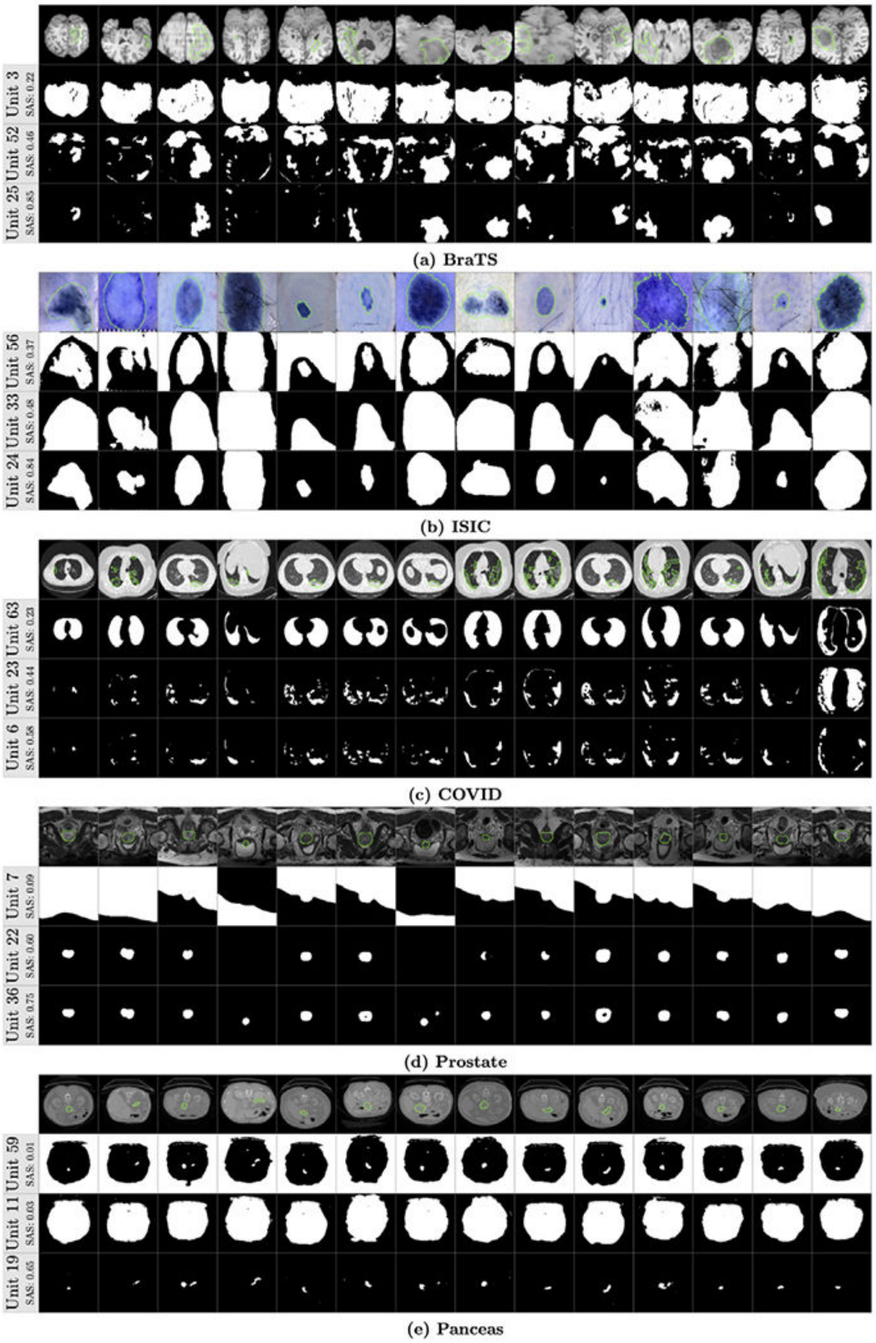


Fig. 8.

Several examples of the segmentation maps from different units. The green contours denote the ground-truth and the SAS is the SA score of each unit.

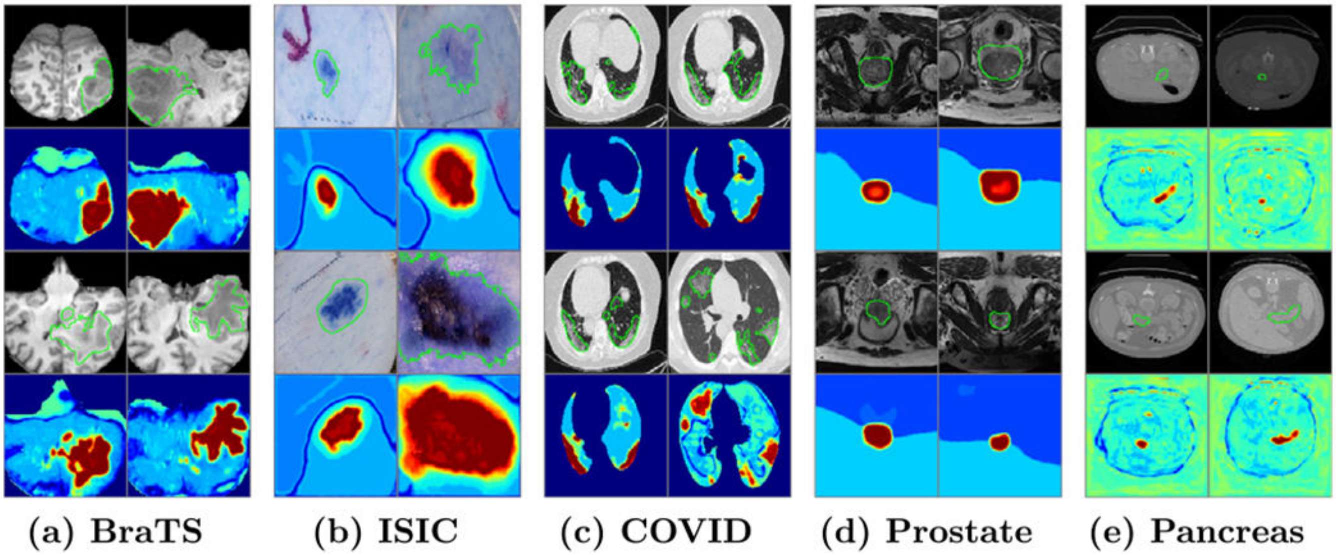
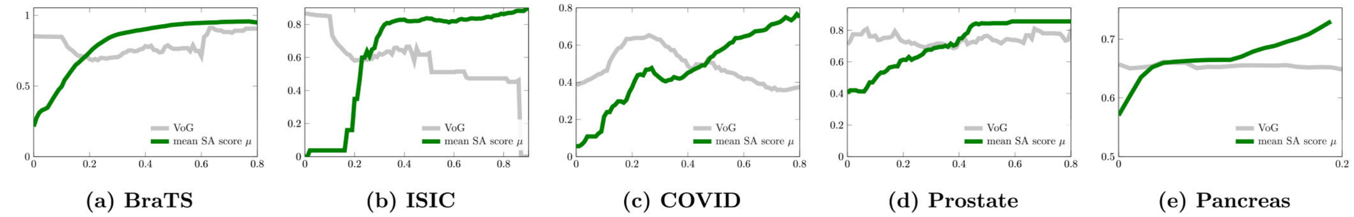


Fig. 9.

The heat-map of the average of unit SAMs on the last layer of U-Net. The first and third rows show the input Images with ground-truth (green contour), the second and the fourth rows show the corresponding heat-map.



(a) **BraTS**

(b) **ISIC**

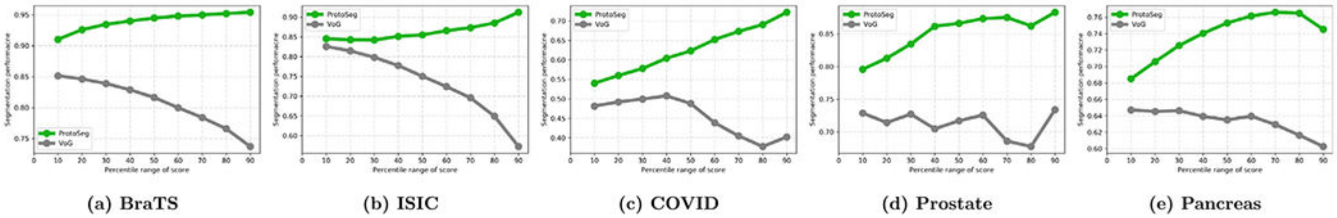
(c) **COVID**

(d) **Prostate**

(e) **Pancreas**

Fig. 10.

The relationship between the Dice accuracy and scores computed by the proposed method (the mean SA score μ, the green lines) and the VoG score (the gray lines) across subjects in each dataset. The y axis is the Dice accuracy of the neural network's output B and the x axis is the mean SA score μ and the scaled VoG score for improving the visualization.



(a) BraTS (b) ISIC (c) COVID (d) Prostate (e) Pancreas

Fig. 11.

The segmentation performance (y-axis) for the test images without ground-truth thresholded by the percentile of the mean SA score μ (green lines) and the VoG score (gray lines) (x-axis). Across the five datasets, the segmentation accuracy increases with an increase in the mean SA scores but not the VoG scores.

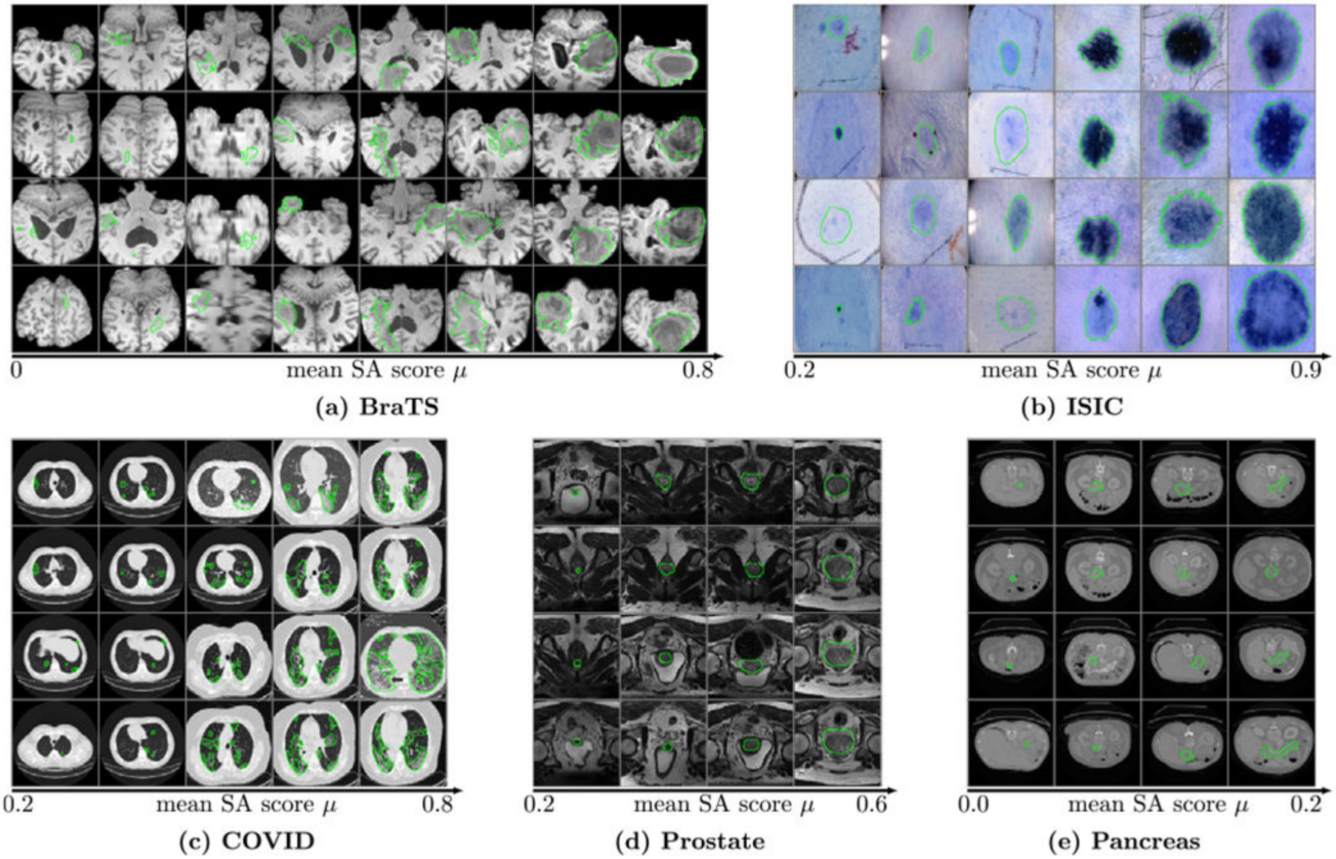


Fig. 12.

The ranking list of images based on the mean SA score μ on the five datasets. The segmentation maps of the images with a low mean SA score μ need more attention.

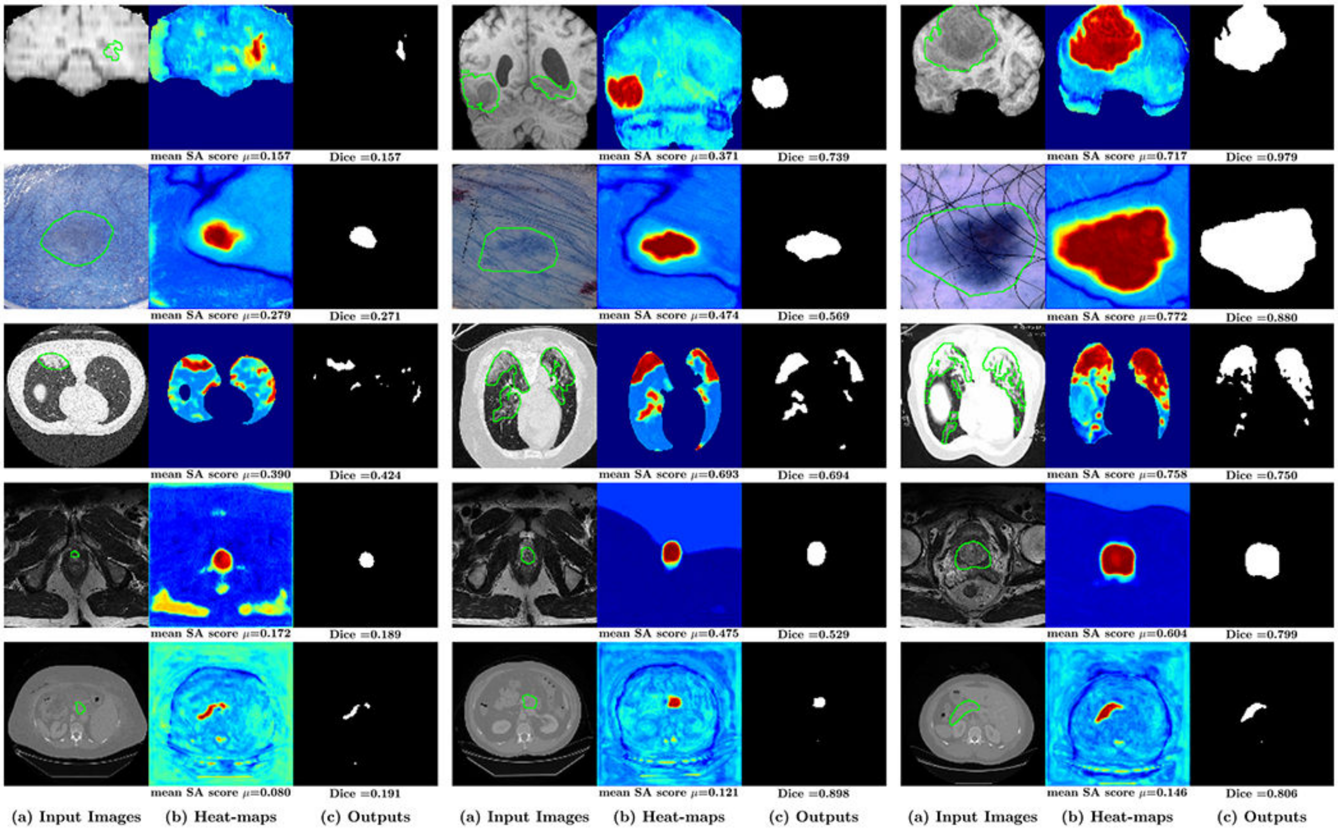


Fig. 13.

Examples of the (a) input images (with ground-truth highlighted with green contours), the corresponding (b) heat-maps computed based on the segmentation of units on last two years and (c) outputs of neural network. The corresponding mean SA score and Dice score are shown on the bottom of each image.

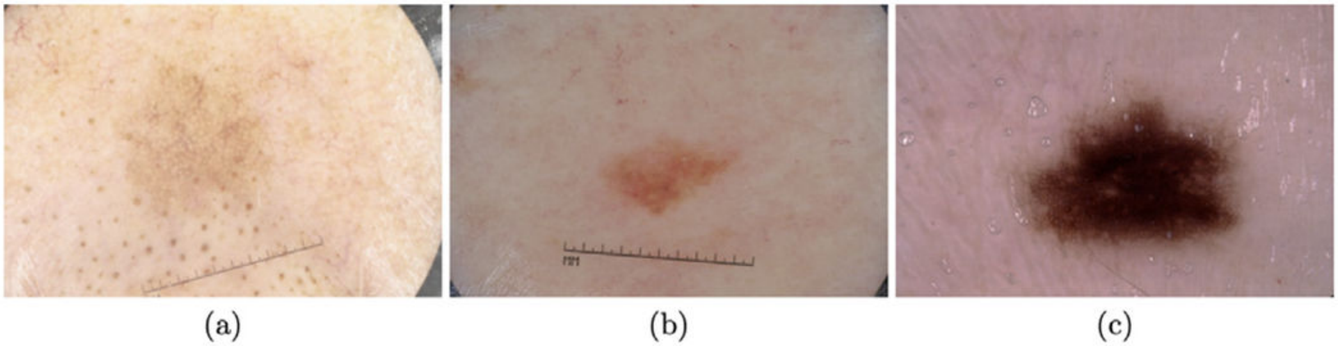
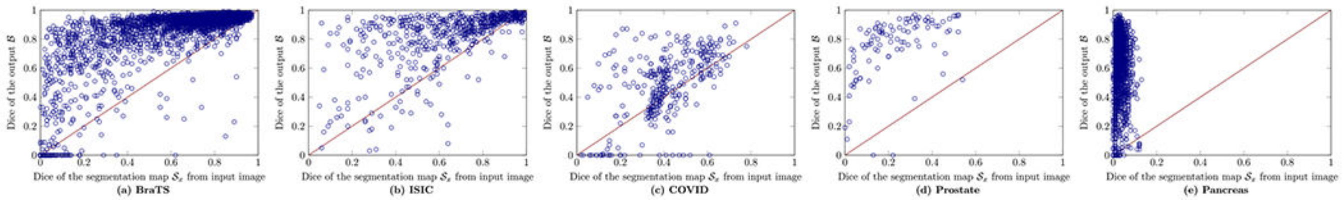


Fig. 14.

Examples of different input images on the ISIC dataset. The separableness is: (a) < (b) < (c).



(a) BraTS

(b) ISIC

(c) COVID

(d) Prostate

(e) Pancreas

Fig. 15.

The scatter plot of the Dice score of the neural network versus the Dice score of segmentation map on the Input Image on different datasets.

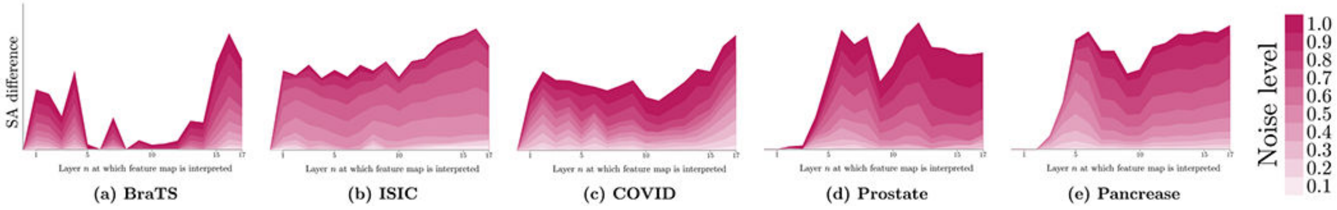


Fig. 16.

The segmentation ability (SA) differences on different layers of U-Net between the noise and the clean inputs. The different noise levels are indicated by different colors.

Table 1

The performance of different methods on five datasets. Output $\mathscr{B}$: the output of the U-Net. Segmentation $S_{f_{18}}$ the SAM on the last layer (the 18th layer in U-Net). It shows that training the network with the SA score in the loss does not hurt the performance.

Dataset	**Offline ProtoSeg**		**Online ProtoSeg**	
	Output $\mathscr{B}$	**Segmentation $S_{f_{18}}$**	**Output $\mathscr{B}$**	**Segmentation $S_{f_{18}}$**
BraTS	0.8471 ± 0.23	0.7944 ± 0.25	0.8428 ± 0.24	**0.8514** ± 0.23
ISIC	0.8330 ± 0.18	0.8260 ± 0.17	0.8353 ± 0.17	**0.8445** ± 0.16
COVID	0.4862 ± 0.22	0.5123 ± 0.21	0.4840 ± 0.22	**0.5597** ± 0.17
Prostate	0.7232 ± 0.28	0.6365 ± 0.34	0.7557 ± 0.20	**0.7585** ± 0.21
Pancreas	0.6475 ± 0.25	0.3439 ± 0.22	0.6441 ± 0.23	**0.6635** ± 0.22

Table 2

The performance of segmentation with sample coverage rates of 100%, 90%, 75% and 50%.

Coverage	100%	90%	70%	50%
BraTS	0.8428 ± 0.24	0.9102 ± 0.11	0.9344 ± 0.06	0.9446 ± 0.04
ISIC	0.8353 ± 0.17	0.8457 ± 0.16	0.8421 ± 0.16	0.8551 ± 0.15
COVID	0.4880 ± 0.22	0.5400 ± 0.17	0.5778 ± 0.15	0.6229 ± 0.13
Prostate	0.7557 ± 0.20	0.7959 ± 0.16	0.8344 ± 0.14	0.8660 ± 0.08
Pancreas	0.6441 ± 0.23	0.6850 ± 0.19	0.7255 ± 0.16	0.7529 ± 0.15

Table 3

The average of the distance between the Dice scores of network output $\mathscr{B}$ and segmentation map $\mathcal{S}_X$ based on colors/intensities of input image.

Dataset	S_x	$\mathscr{B}$	$m(d)$
BraTS	0.657	0.842	0.185
ISIC	0.667	0.835	0.168
COVID	0.411	0.484	0.072
Prostate	0.230	0.757	0.525
Pancreas	0.030	0.644	0.613