Multi-organ Segmentation over Partially Labeled Datasets with Multi-scale Feature Abstraction

Xi Fang, Pingkun Yan, Senior Member, IEEE

Abstract—This paper presents a unified training strategy that enables a novel multi-scale deep neural network to be trained on multiple partially labeled datasets for multi-organ segmentation. Multi-scale contextual information is effective for pixel-level label prediction, i.e., image segmentation. However, such important information is only partially exploited by the existing methods. In this paper, we propose a new network architecture for multi-scale feature abstraction, which integrates pyramid feature analysis into an image segmentation model. To bridge the semantic gap caused by directly merging features from different scales, an equal convolutional depth mechanism is proposed. In addition, we develop a deep supervision mechanism for refining outputs in different scales. To fully leverage the segmentation features from different scales, we design an adaptive weighting layer to fuse the outputs in an automatic fashion. All these features together integrate into a pyramid-input pyramid-output network for efficient feature extraction. Last but not least, to alleviate the hunger for fully annotated data in training deep segmentation models, a unified training strategy is proposed to train one segmentation model on multiple partially labeled datasets for multi-organ segmentation with a novel target adaptive loss. Our proposed method was evaluated on four publicly available datasets, including BTCV, LiTS, KiTS and Spleen, where very promising performance has been achieved. The source code of this work is publicly shared at https://github.com/DIAL-RPI/PIPO-FAN for others to easily reproduce the work and build their own models with the introduced mechanisms.

Index Terms—Medical image segmentation, Multi-scale feature, deep learning, Convolutional neural networks, Multi-organ segmentation, Multiple datasets

I. INTRODUCTION

Automatic multi-organ segmentation, an essential component of medical image analysis, plays an important role in computer-aided diagnosis. For example, locating and segmenting the abdominal anatomy of CT images can be very helpful in cancer diagnosis and treatment [1]. With the surge of deep learning in the past several years, many deep convolutional neural network (CNN) based methods have been proposed and applied to medical image segmentation [2]–[5]. Two main strategies to improve image segmentation performance are: (i) Designing better model architectures and (ii) Learning with larger scale of labeled data.

The state-of-the-art models in medical image segmentation are variants of the encoder-decoder architecture, such as fully convolutional network (FCN) [6] and U-Net [7]. A major focus of the FCN based segmentation methods has been on network structure engineering by incorporating multi-scale features. That is because multi-scale features contain detailed texture information combined with contextual information, which are beneficial for semantic image segmentation. The existing deep learning image segmentation methods that exploit multi-scale features can be generally fulfilled by integrating CNN with skip connections and pyramid parsing modules. Networks employing skip connections to exploit features from different levels are referred as skip-nets. Features in skip-net are multi-scale in nature due to the increasing size of receptive field. U-Net [7] is a typical skip-net and also a strong baseline network to learn pixel-wise information in medical image segmentation. Many works improves segmentation performance by incorporating the latest skip connections such as residual connections [8] and densely connections [9] to segmentation network structure. For instance, Han [10] won ISBI 2017 LiTS Challenge by replacing the convolutional layers in U-net with residual blocks from ResNet. Li et al. [2] replace the encoder part of U-net with DenseNet-169 and obtained a high accuracy of Dice 95.3% on liver CT segmentation.

In addition to exploring new skip connections for efficiently abstracting high level features, incorporating pyramid parsing module for pyramid image feature extraction in FCN has helped to utilize multi-scale information in segmentation tasks [11]–[13]. Pyramid parsing module is designed for extracting same level features at each scales of inputs or intermediate features. A general illustration of pyramid methods is given in Fig. 1. When pyramid parsing module is applied on inputs, multi-scale features are directly extracted from pyramid input images through parallel convolutional channels [11], [14]. The

* Asterisk indicates the corresponding author.

X. Fang and *P. Yan are with the Department of Biomedical Engineering and the Center for Biotechnology and Interdisciplinary Studies at Rensselaer Polytechnic Institute, Troy, NY, USA 12180. (e-mail: fangx2@rpi.edu, yanp2@rpi.edu).

1https://competitions.codalab.org/competitions/15595
other group of methods perform pyramid parsing in one layer for features extracted by a CNN for further abstraction. In those works, features from different scales are only combined at very late stage of the networks to generate final output labels. However, we hypothesize that extracting and maintaining multi-scale features from input-base can efficiently glean hierarchical contextual information and significantly improve the segmentation performance. U-Net is typically a skip-net of pyramid shape that extract pyramid features hierarchically. For this reason it’s intuitive to make use of pyramid features in U-Net architecture for pyramid parsing.

To condense the multi-scale features, we design a new network architecture, which takes pyramid inputs with dedicated convolutional paths to combine features from different scales to utilize the hierarchical information. The hierarchical convolutions through different scales alleviate the semantic gaps between ends of connections. Different from the previous multi-scale mechanism, which extracts multi-scale features through either pyramid parsing module or skip connections between different level features, our network itself can fully utilize the multi-scale contextual feature as shown in Fig. 1b. To fuse the segmentation features from different scales, we further design an adaptive weight layer. In particular, this layer uses an attention mechanism to compute the importance of the features from each scale. The proposed method is thus coined as Pyramid Input and Pyramid Output (PIPO) Feature Abstraction Network (FAN). The proposed method can be easily integrated into the existing U-shape networks to improve the feature representation power of the models.

Deep CNNs have shown great performance for single organ segmentation. However, we often face the problem of multi-organ segmentation in clinical applications. Segmenting multiple organs independently using single organ segmentation algorithms may be straightforward, which, however, loses the holistic view of the image. Therefore, the segmentation performance may be degraded. However, collecting multi-organ annotations for training algorithms is more difficult than annotating single organ datasets. Ideally, researchers could use similar datasets created by different organizations for their research. However, in reality, there will always be some differences that the datasets cannot be directly used, since those data were collected for various purposes. For instance, there are several abdominal CT datasets are publicly available but they are annotated with different targeted organs at risk as shown in Fig. 2. One CT dataset contains the labeled segmentation of the spleen, while another dataset include only the liver annotation. It will be highly advantageous if we can utilize all those datasets together to train a multi-organ segmentation network. In order to achieve this goal, in this paper, we propose a unified training strategy with a novel target adaptive loss.

In this paper, we extensively evaluated the proposed method on the BTCV (Beyond the Cranial Vault) segmentation challenge dataset, MICCAI 2019 Kidney Tumor segmentation (KiTS) Challenge dataset, and the spleen segmentation dataset. We demonstrate very promising performance of medical image segmentation on these datasets compared to the state-of-the-art approaches.

Our contributions in this work can be summarized as follows.

1) A new pyramid-input and pyramid-output network is introduced to condense multi-scale features to reduce the semantic gaps between features from different scales.
2) An image context based adaptive weight layer is used to fuse the segmentation features from multiple scales.
3) A target adaptive loss is integrated with a unified training strategy to enable image segmentation over multiple partially labeled datasets with a single model.
4) Very competitive performance with state-of-the-arts has been achieved by using the developed network on multiple publicly available datasets.

II. RELATED WORKS

A. Multi-scale feature learning

Multi-scale features contain detailed texture and context information, which is helpful for many computer vision tasks including medical image segmentation. Multi-scale feature learning can be generally grouped into two categories. The first type, sometimes referred to as skip-net, combines different level features with skip connections. For example, FPN [18], U-Net [7] and FED-Net [19] use an encoder that gradually down-samples to capture more context, followed by a decoder that learns to upsample the segmentation. Low-level fine appearance information is fused into coarse high-level features through skip connections or convolutional blocks between shallow and deep layers. These works effectively fuse multi-scale context using skip connections, but in the same time introduces huge semantic gap between features at two ends of the connections. UNet++ [20] tries to bridge the semantic gap of skip-net by redesigning the skip-pathways to fuse semantic similar features.

The second type of methods uses pyramid parsing module to extract multi-scale features in the same convolutional level with either pyramid input analysis (P-IA) or pyramid feature analysis (P-FA) as shown in Fig. 1. These features have different effective receptive fields and are concatenated or summed to boost feature representation ability of context information. For example, P-FA methods like PSP-Net [12] apply spatial pyramid pooling to convolutional feature maps for pyramid feature analysis. DeepLab [21] and CE-Net [22] use parallel atrous convolution with different sampling rates to extract multi-scale features to augment segmentation. Qin et al. [23] integrates attention module into the pyramid parsing layer to adapt the network’s receptive field in a data-driven manner. P-IA methods, on the other hand, perform feature analysis from input images with different sizes, i.e. create image pyramid at multiple scales. For example, Farabet et al. [24] enforce scale invariance by applying shared network on different scales of

---

1https://www.synapse.org/#!Synapse:syn3193805/wiki/89480
2https://competitions.codalab.org/competitions/17094
3https://kits19.grand-challenge.org/
Fig. 2. Sample images from the datasets of LiTS, KiTS, Spleen, and BTCV. Some organs are included in all the datasets but only annotated in no more than two of them.

a Laplacian pyramid version of the input image. Kamnitsas et al. [14] employ a dual pathway architecture that processes the input images at multiple scales simultaneously to extract pyramid features to strengthen the feature representation. Since both the pyramid parsing module and skip-net can extract multi-scale context information to help image segmentation, they may also be combined to further boost the performance. Some recent works like [25], [26] integrate features from pyramid input images to the U-Net structure. However, since those features are at different semantic abstraction levels, fusing those features from different scales may cause the problem of semantic gap. Thus, those networks fail to mitigate the multi-scale context information, by only partially utilizing the pyramid shape of U-Net.

B. Segmentation over multiple datasets

Various datasets for semantic segmentation have been presented. Training one general model over multiple datasets will make the features more robust and accurate. When annotations for different datasets are same labels [27], model can be directly trained over these datasets [28]. However, in most cases, different datasets have different annotations. Although many datasets share similar appearance information, annotations difference make it a challenging problem of model generalization over multiple datasets. Many works have been done to deal with the diversity problem. These works can be divided according to different types of annotation difference. When strongly label (pixel-wise annotations) and weak labels (image category, bounding box) exists over multiple datasets. It’s a problem of semi-supervised segmentation. Features are extracted from the encoder are used to learn through multi-task learning. Hong et.al. [29] train class labels and segmentations together at two branches. Papandreou et.al. [30] develop Expectation-Maximization (EM) methods for semantic image segmentation model training on few strongly labeled and many weaklylabeled images, sourced from one or multiple datasets. It’s a problem of semi-supervised segmentation. Features are extracted from the encoder are used to learn through multi-task learning. Hong et.al. [29] train class labels and segmentations together at two branches. Papandreou et.al. [30] develop Expectation-Maximization (EM) methods for semantic image segmentation model training on few strongly labeled and many weaklylabeled images, sourced from one or multiple datasets. When the labels are different but the types of annotations are pixel-wise. Only a few works have been done to solve the problem. Some works [31], [32] design hierarchical classifier for multiple heterogeneous datasets. Each classifier classifies the children labels of a node and the whole classifier is trained. However, semantic hierarchy of the labels is required. Unlike these methods, our proposed approach allows a single model using partial labels, which exploit label proportion information. To train our model, we introduce a new loss function that adapts it-self to the proportion of known labels per example.

C. Multi-organ Segmentation

Accurate and robust segmentation of multiple organs is essential. Three methodologies, statistical models [33], [34], multi-atlas methods [35], [36] and registration-free methods [37], [38] are always used to do multi-organ segmentation. However, these methods are always organ-specific and require prior professional knowledge and manual designing. Recent advances in deep learning and data availability enabled the training of more complex registration-free methods, eg. deep CNN, which neither explicit anatomical correspondences nor hand crafted features. Many studies based on deep CNNs focused on single organ segmentation, particularly for abdominal regions due to the similar intensity and size variation between different target organs. Multi-organ segmentation in abdominal CT has been an important problem to solve for precise diagnosis and treatment. Deep learning based segmentation methods have been developed to segment multiple abdominal organs [28], [39]–[42]. Some works use two-step segmentation to exploit the prior anatomical information [40]–[42]. Specifically, Chen et al. [40] use an organ attention module to guide the fine segmentation. However, majority of the existing works are mostly customized for a certain dataset labeled with all target organs. The lack of variety and quality of datasets make segmentation models specific to particular diseases and also hard to train.

Relaxing the learning requirements to exploit all the available labels open better opportunities for creating large-scale datasets for training deep neural networks. A promising strategy is to use partial annotations from multiple available datasets, which may shared some labeled organs in common. Partial labels are recently introduced to improve image classification performance [43]. We suppose that different datasets are
complementary, which can be used together to train a unified segmentation model without harming the performance.

III. PYRAMID INPUT AND PYRAMID OUTPUT FEATURE ABSTRACTION NETWORK

In this section, we present a novel Pyramid Input and Pyramid Output Feature Abstraction Network (PIPO-FAN), which fully fuses multi-scale context information and semantic similar features with one single network. Pyramid input analysis and pyramid feature analysis are integrated in the proposed network. Our hypothesis is that the semantic information in various depths can be further enhanced by utilizing hierarchical contextual features. PIPO-FAN aims to effectively extract multi-scale features for medical image segmentation, on top of the multi-scale nature of U-net. Fig. 3 shows the overall structure of the proposed PIPO-FAN. The network performs spatial pyramid pooling on input and hierarchical abstract multi-scale features at each level enforced by deep supervision mechanism.

A. Pyramid Input with Equal Convolutional Depth

To seek for patterns from images in different scales, i.e. scale invariance, the proposed network first performs pyramid analysis to the input image by using spatial pyramid pooling and shared convolution to obtain context information in different scales. Unlike the classical U-net based methods, where the scale only reduces when the convolutional depth increases, PIPO-FAN has multi-scale features at each depth and therefore both global and local context information can be integrated to augment the extracted features. After going through one or more convolutional layers, the features are fused together to have hierarchical structural information.

A notable character of PIPO-FAN is that features fused at each level all went through the same number of convolutional layers, i.e. they have equal convolutional depth (ECD). It is achieved by inserting the ResBlocks to the networks, as shown in Fig. 3 by the light blue color boxes. Unlike the existing works, e.g. [25], [26], where features after various depths of convolutions are directly fused together, we deal with the problem of semantic gap using ECD. With the proposed ECD connections, all the fused features at each step are at the same semantic abstraction level to better exploit the pyramid shape of U-Net.

B. Pyramid Output

Furthermore, inspired by the work of deep supervision [44], we introduce deep pyramid supervision (DPS) to the decoding path for generating and supervising outputs of different scales. During the training process, we perform spatial pyramid pooling to the ground truth segmentation to generate labels in all output scales. The training loss is computed by using the corresponding output and ground truth segmentation at the same scale. Weighted cross entropy is used as the loss function in our work, which is defined as

$$L_{DPS} = -\frac{1}{S} \sum_{s=1}^{S} \frac{1}{N_s} \sum_{i=1}^{N_s} \sum_{c=0}^{1} w_{i,s} y_{i,s} c \log p_{i,s} c,$$  (1)
where $p_{c,i,s}^c$ denotes the predicted probability of voxel $i$ belonging to class $c$ (background or liver) in scale $s$

$$p_{c,i,s}^c = \frac{\exp(f_{c,i,s}^c)}{\sum_j \exp(f_{c,j,s}^c)},$$

$y_{c,i,s}$ is the ground truth label in scale $s$, $N_s$ denotes the number of voxels in the scale $s$, and $w_{c,i}^c$ is a weighting parameter for class $c$. DPS can help relieve the problem of gradient vanishing in deep neural networks and learn deep level features with hierarchical contexts. It also enforces the outputs in all scales to maintain structural information.

C. Adaptive Fusion

With the above mentioned DPS mechanism, informative segmentation features are obtained at different scales. Since they may contain complementary context information, we are motivated to fuse these features together to achieve more accurate segmentation. To effectively exploit the contextual information in different scales, we design an adaptive fusion (AF) module to learn the relative importance of each scale and fuse the score maps (i.e., last layer output before softmax) in an automatic fashion.

In particular, this AF module uses an attention mechanism to indicate the importance at each scale. As shown in Fig. 4 after hierarchical abstraction from pyramid input, pyramid output features are propagated into the attention module. To leverage the similar structural information at each scale, Pyramid outputs are first passed into a shared convolutional block to achieve scale invariance.

$$F_s = \text{Conv}(f_s),$$

Those features are then squeezed into a single channel feature vector, which denotes the overall score of output at each scale. Global average pooling (GAP) and global max pooling (GMP) can extract global score of each scale.

$$S_s = \text{GAP}(F_s) + \text{GMP}(F_s),$$

In this work, the sum of them is applied to extract the global score of each scale. $S$ denotes score and $s$ denotes the scale. The values from different scales are then concatenated to feed into a softmax layer to get the corresponding weight for each scale. The weights $w_s$ reflects the importance of feature at scale $s$. After resampling to the original image size, the pyramid output features are summed to get the fused features, where the scale weights are computed as

$$W_s = \frac{\exp(S_s)}{\sum_s \exp(S_s)},$$

Another softmax layer is applied to obtain the final segmentation result

$$p_{c,a}^c = \frac{\exp(\sum_{s=1}^S w_sf_s)}{\sum_c \exp(\sum_{s=1}^S w_sf_s)}.$$
Considering the partial labels are subset of all labels, probabilities can be merged according to the known labels. Therefore, we treat the unknown labels as background to allow computing the loss. The TAL function is defined as

\[ L_{TAL} = \mathbb{1}_{[c_i^2=1]} \log p_i^{c_1} + \mathbb{1}_{[c_i^2=0]} \log(1 - p_i^{c_1}), \]  

where \(c_1\) denotes the labeled organ.

TAL can be easily implemented by modifying the last layer of the segmentation network to have multiple branches to segment all organs labeled in these datasets. After the second softmax layer, each branch gets a probability of that class. When training on the partially labeled datasets, probabilities of labeled target organs are preserved, while other probabilities are merged as “non-target” class in that dataset. For example, to train on LiTS dataset, we preserve the probability of liver and combine the other branches probabilities to the probability of non-liver. Then a binary cross entropy can be computed. Similarly, training on the KiTS and spleen datasets can be completed. Such a mechanism allows gradients to be back propagated through the branches with the labeled organs. A general segmentation model can then be obtained, which is able to segment multiple organs when they are present in an image.

IV. EXPERIMENTS

A. Materials

We evaluated our model on four publicly available datasets, LiTS (Liver tumor segmentation challenge) [45], KiTS (Kidney tumor segmentation challenge) [46], the Spleen segmentation dataset [15] and BTCV (Beyond the Cranial Vault) segmentation challenge dataset [47]. The first three datasets are single-organ annotation datasets and the last one has multiple organs annotated. Fig. 2 provides a summary of the available organ annotations in those datasets.

LiTS consists of 131 training and 70 test images. The data were collected from different hospitals and the intra-slice resolutions of the CT scans vary between 0.45mm and 6mm and the inter-slice resolutions vary between 0.6mm and 1.0mm for inter-slices. KiTS consists of 210 training and 90 test images collected from 300 patients, who underwent nephrectomy for kidney tumors. The spleen segmentation dataset is composed of patients undergoing chemotherapy for liver metastases at Memorial Sloan Kettering Cancer Center (New York, NY, USA) [15]. The BTCV segmentation challenge dataset contains 47 subjects with segmentation of all abdominal organs except duodenum. 30 of them have labels of both two kidneys. We randomly split those subjects into 21 for training and 9 for validation. We select three abdominal organs in partially labeled datasets as the target organs.

In these datasets, the size of each slice is 512×512 pixels. To speed up the model training, we resized the axial slices into 256×256 pixels, where the boundary information is still well preserved. We keep the CT imaging HU values using in range of [-200, 200] to have a better contrast on the abdominal organs. For each epoch, we randomly select three continuous slices containing target organ label from all the CT training volumes and crop a patch with size of 224×224 as input to the network. After obtaining the segmentation volume, the connected component analysis was performed to keep only the largest component as the segmentation result of liver and spleen, and the largest two components as the segmentations of kidney.

B. Implementation Details

Our implementation is based on the open-source platform PyTorch [48]. All the convolutional operations are followed by batch normalization and ReLU activation. For network training, we use the RMSprop optimizer. For multi-organ segmentation, we set learning rate to be 0.0002 and the maximum number of training epochs to be 4000. For single organ segmentation, we set the initial learning rate to be 0.002 and the maximum number of training epochs to be 2500. The learning rate decays by 0.01 after every 40 epochs. For the first 2000 epochs, the deep supervision losses are applied to focus on the feature abstraction ability in each scale. For the remaining epochs, the adaptive weighting layer is activated and the deep supervision loss stops, which helps to optimize only the adaptively fused score map output.

When training a single-organ segmentation model, empirically, we set the weights in Eqn. (1) to be 0.2 and 1.2 for background and the organ, respectively, to counteract the unbalanced training samples. When training a multi-organ segmentation model, for simplicity and generalization ability, all the weights in the cross entropy function are set to be 1.

When two or more datasets are used, the model is trained by batch normalization and ReLU activation. For network training, we use the RMSprop optimizer. For multi-organ segmentation, we set learning rate to be 0.0002 and the maximum number of training epochs to be 2500. The learning rate decays by 0.01 after every 40 epochs. For the first 2000 epochs, the deep supervision losses are applied to focus on the feature abstraction ability in each scale. For the remaining epochs, the adaptive weighting layer is activated and the deep supervision loss stops, which helps to optimize only the adaptively fused score map output.

When training a single-organ segmentation model, empirically, we set the weights in Eqn. (1) to be 0.2 and 1.2 for background and the organ, respectively, to counteract the unbalanced training samples. When training a multi-organ segmentation model, for simplicity and generalization ability, all the weights in the cross entropy function are set to be 1.

When two or more datasets are used, the model is trained alternatively on those datasets by using one at a time.

In our work, five-fold cross validation on LiTS and KiTS was employed to evaluate the performance of the models. Dice score is used as the evaluation criterion. When preparing the test result submission to the LiTS challenge websites, we used majority voting to combine the outputs of the five models to get the final segmentation. Our implementation code is open-sourced available at https://github.com/DIAL-RPI/PIPO-FAN.

C. Multi-organ Segmentation

We evaluate our proposed training strategy on BTCV validation dataset. Liver, kidney and spleen are three target abdominal organs to compare the segmentation performance. In Table I different combinations of datasets are used to train the multi-organ segmentation model. The proposed PIPO-FAN is used as the segmentation model. Dice score is used as the evaluation criterion. When training on the BTCV dataset, the model serves as the baseline of multi-organ segmentation.

| Datasets | Liver | Kidney | Spleen | Average |
|----------|-------|--------|--------|---------|
| BTCV     | 95.8  | 92.7   | 92.3   | 93.6    |
| BTCV + LiTS | 95.6  | 91.6   | 95.4   | 94.2    |
| BTCV + KiTS | 94.2  | 91.9   | 94.5   | 93.5    |
| BTCV + Spleen | 95.9  | 93.5   | 93.8   | 94.4    |
| LiTS + KiTS + Spleen | 95.6  | 91.6   | 93.8   | 93.7    |
| All datasets | 95.9  | 91.9   | 95.5   | 94.4    |
Fig. 5. Segmentation examples on BTCV data. Each row from left to right shows the following images in order: (i) input image, (ii) ground truth segmentation, (iii) segmentation results of models trained with BTCV only, (iv) with partially labeled data, and (v) with all the datasets, respectively. Red, blue and green colors depict the segmentation of the liver, the spleen, and the kidney, respectively.

TABLE II

| Architecture       | Liver | Kidney | Spleen | Average |
|--------------------|-------|--------|--------|---------|
| U-Net [7]          | 95.6  | 89.7   | 91.0   | 92.1    |
| ResU-Net [10]      | 95.1  | 91.3   | 90.9   | 92.4    |
| PIPO-FAN (DPS)     | 93.7  | 92.6   | 90.1   | 92.8    |
| PIPO-FAN (DPS + AF)| 95.8  | 92.7   | 92.3   | 93.6    |

Three partially labeled datasets, LiTS, KiTS and Spleen segmentation datasets are individually added as additional training dataset to enhance training procedure. The combination of four datasets are also used for the unified training. Some example results are shown in Fig. 5. We notice that training with the additional partially labeled datasets improves the multi-organ segmentation, especially the spleen dataset. Compared to using BTCV alone, the use of additional datasets significantly boosts the segmentation performance of the spleen, which may be because the spleen has similar appearances across all the datasets.

D. Model Analysis and Ablation Studies

We further compared our proposed PIPO-FAN against several other classical benchmark 2D segmentation networks, including U-Net [7], ResU-Net [10], and DenseU-Net [2], to demonstrate the effectiveness of DPS and AF. For fair comparison, U-Net, ResU-Net and our PIPO-FAN are all 19-layer networks. The segmentation results on BTCV are shown in Table II. Some example results on LiTS are shown in Fig. 6. The DenseU-Net is the 2D DenseU-Net architecture used in [2] and the encoder part is Densenet-169. According to experimental results from [49], adding residual connections makes little difference in performance on the KiTS data. Thus, in our experiments on the KiTS data, we used ConvBlocks in PIPO-FAN rather than ResBlocks. All these 2D networks are trained from scratch in the same environment. We evaluate the performance of the above networks on each dataset through five-fold cross validation. The 131 labeled LiTS data and 210 KiTS data are split into 5 folds, respectively. Each fold is used only once for validation, while the other four are used for training.

The five-fold cross validation results are shown in Table III. The conducted t-test shows that PIPO-FAN significantly outperforms U-Net, ResU-Net and DenseU-Net with p-values of 0.004, 0.025, and 0.002 in LiTS data respectively.

TABLE III

| Architecture       | LiTS       | KiTS       |
|--------------------|------------|------------|
| U-Net [7]          | 93.9 ± 0.50| 95.8 ± 0.91|
| ResU-Net [10]      | 94.1 ± 0.88| 94.8 ± 1.06|
| DenseU-Net [2]     | 94.1 ± 0.30| 94.2 ± 2.08|
| PIPO-FAN (DPS)     | 95.3 ± 0.62| 96.5 ± 0.55|
| PIPO-FAN (DPS + AF)| 95.6 ± 0.48| 96.2 ± 1.02|

In Section III-A, Table IV shows the the segmentation performance under different network configurations. As expected, using PIPO always outperforms single-scale input/output segmentation, which is indeed a ResU-Net. With PIPO to explore
Fig. 6. Segmentation examples of different methods on LiTS data. From left to right are the raw image, results of U-Net, ResU-Net, DenseU-Net and our proposed PIPO-FAN, the red depicts correctly predicted liver segmentation, the blue shows false positive, green shows false negative.

| Architecture                          | Avg. Dice | Glb. Dice |
|---------------------------------------|-----------|-----------|
| Single-scale input/output             | 94.1      | 94.5      |
| PIPO w/o ECD (DPS)                    | 95.1      | 95.2      |
| PIPO w/o ECD (DPS + AF)               | 95.2      | 95.1      |
| PIPO with ECD (DPS)                   | 95.3      | 95.4      |
| PIPO with ECD (DPS + AF)              | **95.6**  | **95.8**  |

TABLE V
PERFORMANCE EVALUATION OF VARYING THE NUMBERS OF THE INPUT AND OUTPUT SCALES ON LiTS DATASET.

| Input scale | Output scale | Avg. Dice (%) |
|-------------|--------------|---------------|
| 1           | 1            | 94.1          |
| 3           | 1            | 94.9          |
| 5           | 1            | 94.5          |
| 3           | 3            | 95.1          |
| 5           | 3            | 95.3          |
| 5           | 5            | **95.6**      |

Multi-scale image information, using ECD results in consistent performance improvement. It is worth noting that using ECD with only deep pyramid supervision (DPS) performs better than using AF without ECD. This not only shows that ECD is effective in extracting image features, but also illustrates the necessity of having good features for adaptive fusion module to work efficiently.

We also tried different number of scales for the input and output to evaluate the relationship between scales and model capacity. The results are shown in Table V. It can be seen that the larger scale numbers of input scales and output scales, the better segmentation accuracy the model obtains. It may be because the higher scales can provide larger receptive fields and thus enhance the contextual information. This in turn helps the feature abstraction, i.e. extracting representative segmentation features. A special case is that when the number of input scale increased from 3 to 5, but the number of output scale remains at 1, the segmentation performance dropped. It may be because increasing the input scale alone without additional output supervision adds difficulty to the network training. Based on the results, we empirically make use of five input and output scales in the final version of our work.

E. Comparison with state-of-the-arts on LiTS challenge

Most of the state-of-the-art methods on liver CT image segmentation takes two steps to complete the task, where a coarse segmentation is used to locate the liver followed by fine segmentation step to obtain the final segmentation [2], [10]. However, such two-step methods can be computationally expensive. For example, the method in [2] takes 21 hours...
to finetune a pretrained 2D DenseUNet and another 9 hours to finetune the H-DenseUNet with two Titan Xp GPUs. In contrast, our proposed method can be trained on a single Titan Xp GPU in 3 hours. More importantly, when segmenting a CT volume, our method only takes 0.04s for one slice on a single GPU, which is, to the best of our knowledge, the fastest segmentation method compared to other reported methods. In the same, we are able to obtain the same Dice performance and even better symmetric surface distance (SSD) (ASSD: 1.413 < 1.450, MSSD: 24.408 < 27.118, 2.421 < 3.150). Table VI shows the performance comparison with other published state-of-the-art methods on LiTS challenge dataset. Despite its simplicity, our proposed 2D network segments the liver in a single step and can obtain a very competitive performance with less than 0.2% drop in Dice, compared to the top performing method – DeepX [51].

V. CONCLUSION

In this paper, we propose a novel network architecture for multi-scale feature abstraction, which incorporates multi-scale features in a hierarchical fashion at various depths for medical image segmentation. The proposed 2D network with only a single step shows very competitive performance compared with other multi-step 3D networks in CT image segmentation. We further develop a unified segmentation strategy to train the proposed network on multiple partially labeled datasets for multi-organ segmentation. The new strategy gives the segmentation network better robustness and accuracy by enlarging the training dataset. The source code of our work has been open sourced to enable further testing and development in a larger scale on other imaging modalities.

ACKNOWLEDGMENT

The authors would like to thank NVIDIA Corporation for the donation of two Titan Xp GPUs used for this research. We would also like to thank Prof. George Xu (RI), Mr. Zhao Peng (USTC), and Dr. Sheng Xu (NIH) for the insightful discussions.

REFERENCES

[1] R. Wolz, C. Chu, K. Misawa, M. Fujiwara, K. Mori, and D. Rueckert, “Automated abdominal multi-organ segmentation with subject-specific atlas generation,” IEEE Transactions on Medical Imaging, vol. 32, no. 9, pp. 1723–1730, Sep. 2013.
[2] X. Li, H. Chen, X. Qi, Q. Dou, C. Fu, and P. Heng, “H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from CT volumes,” IEEE Transactions on Medical Imaging, vol. 37, no. 12, pp. 2663–2674, Dec 2018.
[3] Q. Zhu, B. Du, and P. Yan, “Boundary-weighted domain adaptive neural network for prostate mr image segmentation,” IEEE Transactions on Medical Imaging,, pp. 1–1, 2019.
[4] K. He, X. Cao, X. Shi, D. Nie, Y. Gao, and D. Shen, “Pelvic organ segmentation using distinctive curve guided fully convolutional networks,” IEEE Transactions on Medical Imaging, vol. 38, no. 2, pp. 585–595, Feb 2019.
[5] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, “Drinnet for medical image segmentation,” IEEE Transactions on Medical Imaging, vol. 37, no. 11, pp. 2453–2462, Nov 2018.
[6] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in CVPR, 2015, pp. 3431–3440. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7298965
[7] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in MICCAI, 2015, pp. 234–241.
[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in CVPR, 2016, pp. 770–778.
[9] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in CVPR, 2017, pp. 2261–2269.
[10] I. Ioffe, “Automatic Liver Lesion Segmentation Using A Deep Convolutional Neural Network Method,” arXiv:1704.07239, Apr. 2017. [Online]. Available: http://arxiv.org/abs/1704.07239
[11] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, “Efficient piecewise training of deep structured models for semantic segmentation,” in CVPR, 2016, pp. 3194–3203.
[12] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in CVPR. IEEE Computer Society, July 2017, pp. 6230–6239. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.660
[13] X. Fang, B. Du, S. Xu, B. J. Wood, and P. Yan, “Unified multi-scale feature abstraction for medical image segmentation,” arXiv preprint arXiv:1910.11456, 2019.
[14] K. Kannitissas, C. Legid, V. E. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon et al., “Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation,” Medical Image Analysis, vol. 36, pp. 61–78, 2017.
[15] A. L. Simpson, J. N. Leal, A. Pugalenthii, P. J. Allen, R. P. DeMatteo, Y. Fong et al., “Chemotherapy-induced splenic volume increase is independently associated with major complications after hepatic resection for metastatic colorectal cancer,” Journal of the American College of Surgeons, vol. 220, no. 3, pp. 271–280, 2015.
[16] X. Chen, R. Zhang, and P. Yan, “Feature fusion encoder decoder network for automatic liver lesion segmentation,” in 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), April 2019, pp. 430–433.
[17] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, “Attention to scale: Scale-aware semantic image segmentation,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3640–3649, 2015.
[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
[19] O. Oktay, J. Schlemper, L. L. Folgoe, M. Lee, M. Heinrich, K. Misawa et al., “Attention u-net: Learning where to look for the pancreas,” arXiv preprint arXiv:1804.03999, 2018.
[20] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” in DL in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, 2018, pp. 3–11.
[21] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” IEEE Trans. PAMI, vol. 40, no. 4, pp. 834–848, 2018. [Online]. Available: https://doi.org/10.1109/TPAMI.2017.2699184
[22] Z. Gu, J. Cheng, H. Fu, K. Zhou, H. Hao, Y. Zhao et al., “Ce-net: Context encoder network for 2d medical image segmentation,” IEEE Transactions on Medical Imaging, vol. 38, no. 10, pp. 2281–2292, Oct 2019.
[23] Y. Qin, K. Kannitissas, A. Sanga, J. Nanavati, G. Cottrell, A. Criminisi et al., “Autofocus layer for semantic segmentation,” in International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), September 2018, pp. 3–11.
[24] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1915–1929, Aug 2013.
[25] S. E. A. Raza, L. Cheung, D. Epstein, S. Pelengaris, M. Khan, and N. M. Rajpoot, “MIMO-Net: A multi-input multi-output convolutional neural network for cell segmentation in fluorescence microscopy images,” in IEEE, 14th International Symposium on Biomedical Imaging (ISBI), April 2017, pp. 337–340.
[26] H. Fu, J. Cheng, Y. Xu, D. W. K. Wong, J. Liu, and X. Cao, “Joint optic disc and cup segmentation based on multi-label deep network and polar transformation,” IEEE transactions on medical imaging, vol. 37, no. 7, pp. 1597–1605, 2018.
[27] L. Rundo, C. Han, Y. Nagano, J. Zhang, R. Hataya, C. Miliello et al., “USE-Net: incorporating squeeze-and-excitation blocks into u-net for prostate zonal segmentation of multi-institutional mri datasets,” arXiv preprint arXiv:1904.08254, 2019.
