서울대학교 대학원 의과학과

학부연구생인턴 보고서

연 구 기 간: 27기(2021. 6. 21. ~ 2021. 8. 13.)

연 구 제 목: Deep Learning Approaches for Fully Automated Methods for Medical Image Segmentation

성 명: 김혜성 (인 또는 서명)

지도교수 성 명: 이재성 (인 또는 서명)

(※ A4용지 10장 이내, 바탕체, 11포인트, 줄간격 160%. 아래 항목대로 기술하기 힘든 경우에는 해당 항목만을 기술함)

1. Abstract

Development of robust and accurate fully automated methods for medical image segmentation is crucial in clinical practice and radiomics studies. In this work, we proposed different types of automatic segmentation methods for head and neck tumors from PET and CT images based on the convolutional neural networks (CNN). Specifically, we first introduced a multi-channel 3D U-Net to segment the tumor with the concatenated PET and CT images in the context of the MICCAI 2021 Head and Neck Tumor segmentation challenge (HECKTOR). Our models were based on the U-Net architecture with residual layers, supplemented with Squeeze-and-Excitation Normalization and Residual Recurrent Convolutional Neural Network (RRCNN). We tweaked hyperparameters empirically and modified the basic model by adding RRCNN and eliminating the upsampling pathway. Then, we estimated the segmentation results on the train set (537 or 576) with Dice similarity coefficient (DSC), as well as the test set (135 or 96).

2. Introduction (Research Background and Purpose)

Computed tomography (CT) and positron emission tomography (PET), knowing as structural imaging techniques and functional imaging techniques, respectively, are broadly used in clinical practice to provide detailed structural information about human anatomy and pathology detection applications. In clinical practice, PET and CT each provides cellular activity and anatomical features, which are used for radiotherapy treatment planning, initial staging and response assessment. Combining with CT and PET modalities utilize both structural and functional information and provides synergistic information for tumor segmentation.

With that being said, the segmentation of the radiomics workflow is time-consuming bottleneck and has high variability. Radiologists and experts need to annotate each 2D slices images, namely, tedious and manual segmentation. Under these circumstances, a fully automated segmentation is highly recommended to automate the whole process and facilitate its clinical routine process.

The MICCAI 2021 Head and Neck Tumor segmentation challenge (HECKTOR) aims at evaluating automatic algorithms for segmentation of Head and Neck (H&N) tumors in combined PET and CT images. A dataset of 224 patients from five medical centers (CHGJ, CHMR, CHUM, CHUP, CHUS) with histologically proven H&N center in the oropharynx is provided. All images were re-annotated by an expert for the purpose of the challenge in order to determine primary gross tumor volumes (GTV) where the methods are evaluated using the Dice score (DSC), precision and recall.

In this study, we describe our approach based on convolutional neural networks supplemented with Squeeze-and-Excitation Normalization (SE Normalization or SE Norm) layers, Residual Recurrent Convolutional Neural Network (RRCNN), Upsampling pathway, and modifications on hyperparameters.

3. Method

3.1 Data Preprocessing & Sampling

We trained and validated our models on brain PET/CT data of 224 patients retrospectively obtained at 5 medical centers (CHGJ, CHMR, CHUM, CHUP, CHUS). We divided the dataset into two cases in a train set and a validation set. The ratios of the train set and validation set were 4:1 and 6:1 respectively. We unified the ratio to 6:1 in the middle of the experiment to compare network performances to others.

Both PET and CT images were first resampled to a common resolution of 1 x 1 x 1 mm^3. The images were resampled with spline interpolation of degree 3 and the segmentation were resampled by nearest neighbor interpolation. Each training example was a patch of 144 x 144 x 144 voxels, based on the given bounding boxes containing the tumor class provided by organizers, extracted from a whole PET/CT image.

CT intensities were clipped in the range of [-1024, 1024] Hounsfield Units and then mapped to [-1, 1]. PET and CT images were transformed independently with the use of Z-score normalization, performed on each patch.

3.2 Network Architecture

U-Net is one of the CNN models that showed decent performance for image segmentation and denoising and is often used in the medical field.

Basically, U-Net is a model that adds a skip connection to an encoder-decoder structure. U-Net has a contracting path and an expansive path. U-Net networks consisted of convolution layers, rectified linear units (ReLU, an activation function defined as f(x) = max(0,x) and used to provide nonlinearity to the learning model), 2 x 2 max pooling layers, deconvolution layers, and a 1 x 1 convolution layers.

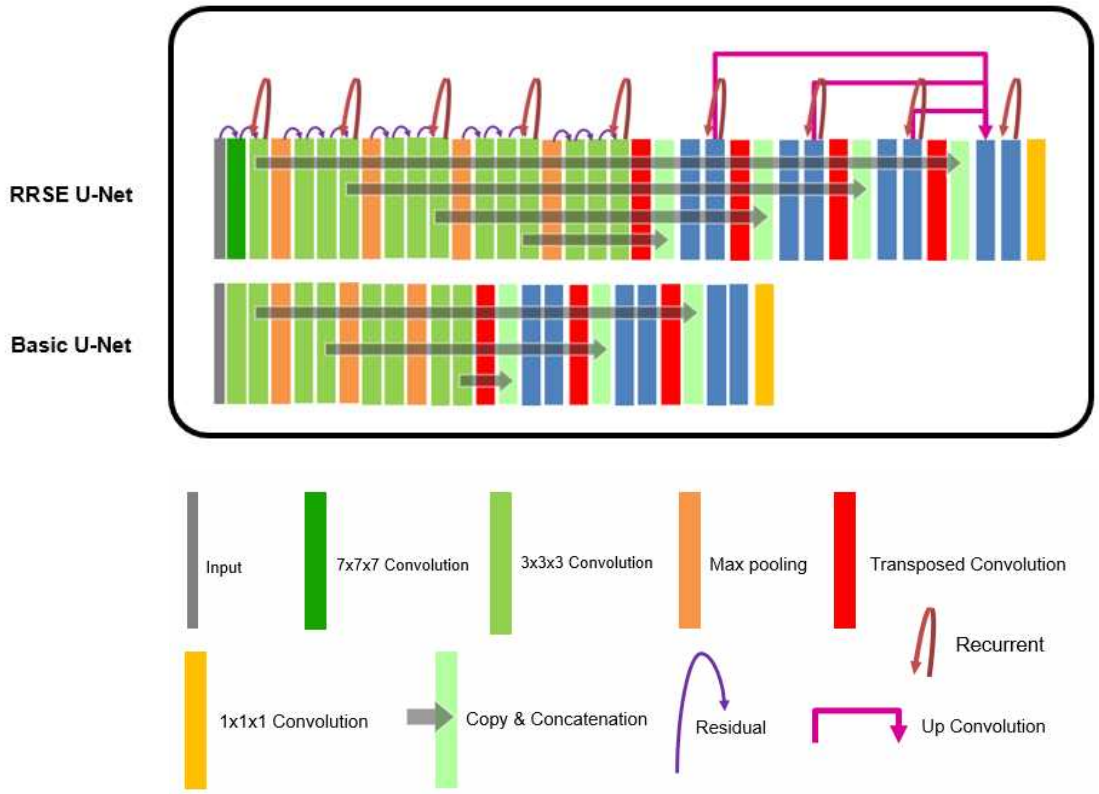

Figure 1. CNN architectures used to learn CT, PET, and Ground Truth. (A) Residual Recurrent SE U-Net (RRSE U-Net). (B) Basic U-Net. Gray stripes at far left indicate inputs to CNN, and yellow stripes at right indicate output. Each box represents multi-channel feature map. Data flow is left to right through contracting path to capture context and symmetric expansive path to recover image. Purple arrows stand for copying feature maps, and light-green boxes are copied feature map. Scarlet arrows indicate recurrent with time stamp 2 and kerel size is 3.

In the contracting path, for the first layer in RRSE U-Net used the 7 x 7 x 7 convolutions and 3 x 3 x 3 convolutions. The 3 x 3 x 3 convolutions and ReLU function are repeated third for each dimension in RRSE U-Net. The 3 x 3 x 3 convolutions and ReLU function are repeated twice for each dimension in Basic U-Net. When downsampling, 2 x 2 x 2 max polling is used with two stride and number of features uses double of the previous value. In the expansive path, the 2 x 2 x 2 up-convolution is applied and the number of features uses half of the previous value. When up-convolution is performed by cropping the feature

map before max-pooling in the contracting path, concatenation is performed. Then 3 x 3 x 3 convolution and ReLU function are repeated third for RRSE U-Net and twice for Basic U-Net.

The main advantage of the U-Net structure is that the skip connection delivered the first convolution of the decoder to the end of the encoder for each dimension. When the image is downsampled and then upsampled, thus the size of the image is reduced and enlarged, informative pixel information vanished. Because U-Net structure consisted of the encoder and decoder, the skip connection can leverage this problem. Important information can be delivered by adding skip connection from the encoder to the decoder which generally gives the more accurate prediction and thus get much clearer image outputs.

We performed the process of image preprocessing and implemented the networks using the Pytorch, an open-source library for deep learning.

3.3 Loss function & Image Analysis

The unweighted sum of the Soft Dice Loss was employed to train the model. The Soft Dice Loss for one training example can be written as

$$L_{Dice}(y,\hat{y}) = 1 - \frac{2\sum_i^{\mathrm{N}} y_i \hat{y}_i + 1}{\sum_i^{\mathrm{N}} y_i + \sum_i^{\mathrm{N}} \hat{y}_i + 1}$$

where, $y_i \in \{0,1\}$ indicates the label for the $i-th$ voxel, $\hat{y}_i \in \{0,1\}$ indicates the predicted probability for the $i-th$ voxel, and N indicates the total numbers of voxels. We add 1 to the numerator and denominator in the Soft Dice Loss to avoid the zero division in cases when the tumor class is not present in training patches.

Dice similarity coefficient measured the overlap of the segmented output derived from CT&PET data and ground truth according to the following equation:

$$DSC = \frac{2 \times N_{(Segmented\ Output \cap Ground\ Truth)}}{N_{Segmented\ Output} + N_{Ground\ Truth}}$$

where $N_{Segmented\ Output}$ and $N_{Ground\ Truth}$ are, respectively, the number of head and neck tumor (or air) voxels derived from the proposed model's segmented output with two input data (CT and PET) and ground truth data. $N_{(Segmented\ Output \cap Ground\ Truth)}$ indicated the number of overlapped voxels between segmented output and ground truth. Segmented output and ground truth were resampled to the dimension is 144 x 144 x 144 and the voxel size is 1 x 1 x 1 mm^3. The segmented output of the proposed model's segmented by the threshold of 0.5. The voxels having a value of more than 0.5 were classified as tumor; those having a value of less than 0.5 were denoted as air.

In boundary detection scenario, the value of $N_{Segmented\ Output}$ and $N_{Ground\ Truth}$ are either 0 or 1, representing whether the pixel is boundary (value of 1) or not (value of 0). Therefore, the denominator is the sum of total boundary pixels of both prediction and ground truth, and the numerator is the sum of correctly predicted boundary pixels because the sum increments only when $N_{Segmented\ Output}$ and $N_{Ground\ Truth}$ match (both of value 1).

3.3 Training Procedure

The models were trained for 100 epochs using Adam optimizer on one GPU NVIDIA GeForce GTX 1080 Ti (11 GB) with a batch size of 1 or 2 (one sample per worker). The cosine annealing schedule was applied to reduce the learning rate from 10^{-3} to 10^{-5} within every 25 epochs.

4. Results

First, we used U-Net architecture with the use of SE Norm layers with different hyperparameters. The mean and standard deviation of each metric were computed across all data samples in the corresponding train set.

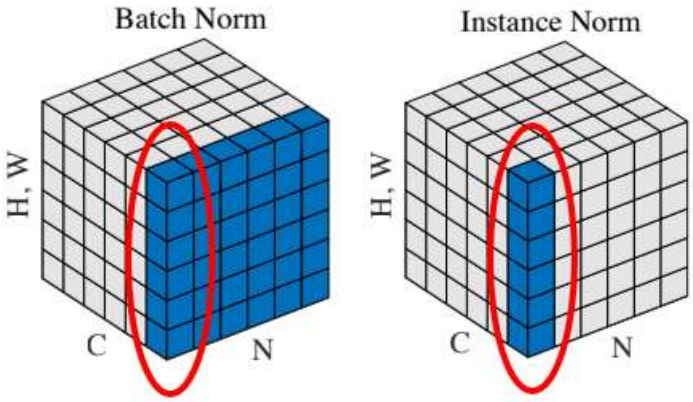

Figure 2: Batch Norm and Instance Norm. H, W stand for Height and Width respectively, C for channels, N for batch size.

Table 1: Comparison of normalization method, InstanceNorm3d and BatchNorm3d. The model used BatchNorm3d with batch size 1 failed to learn images features.

Normalization	Filters	Batch Size	Data Augmentation	DSC
InstanceNorm3d	24	1	RandomRotation Mirroring	0.729
BatchNorm3d	24	1	RandomRotation Mirroring	0.490

Table 2: Comparison of hyperparameters within the model used BatchNorm3d. When it comes to BatchNorm3d, batch size plays an important role in hyperparameters.

Normalization	Filters	Batch Size	Data Augmentation	DSC
BatchNorm3d	14	2	-	0.685
BatchNorm3d	24	1	RandomRotation Mirroring	0.490

Table 3: Comparison of hyperparameters within the models used InstanceNorm3d. When it comes to InstanceNorm3d, the model consists of a high number of filters

and batch size performed better. Furthermore, adding RandomRotation and Mirroring for each iteration leverages the small dataset.

Normalization	Filters	Batch Size	Data Augmentation	DSC
InstanceNorm3d	24	1	RandomRotation Mirroring	0.729
InstanceNorm3d	14	2	RandomRotation Mirroring	0.729
InstanceNorm3d	16	1	-	0.702
InstanceNorm3d	15	2	-	0.708
InstanceNorm3d	14	2	-	0.718
InstanceNorm3d	2	8	-	0.639

Table 4: Comparison of data augmentation when other hyperparameters were fixed. We added random normally distributed noise to input tensor, the maximum variance of added noise with the value of 0.1 and 0.01. There is no significant difference in the results. DSCs are relatively lower when we added RandomNoise. At this time we changed the ratio of train set and validation set to 6:1.

Normalization	Filters	Batch Size	Data Augmentation	DSC
InstanceNorm3d	24	1	RandomRotation Mirroring RandomNoise=0.1	0.695
InstanceNorm3d	24	1	RandomRotation Mirroring RandomNoise=0.01	0.696

Secondly, we transformed the proposed architecture with the use of RRCNN and Upsampling pathway. We set hyperparameters with the full use of GPU. The

baseline model, the encoder consists of 4 stages to downsample image with stride 2 and doubling the feature maps, the decoder consists of 4 stages to upsample image with stride 2 and halve the feature maps and 1 x 1 x 1 convolution to segment input into 2 channels background and foreground by the threshold of 0.5 as the same resolution of the input.

Each RRCNN block has time stamp 2 and added to the last convolutional layer of each dimension. In the first model, RRCNN block in both contracting and expansive path without Upsampling pathway was added. In the second model, RRCNN block was added only in contracting path without Upsampling pathway. Third model has Upsampling pathway with adding RRCNN in contracting path except at the last convolutional layer in the lowest dimension because of the limited GPU usage. Last model has Upsampling pathway and instead of adding RRCNN at the very last layer in each dimension. We replaced each last convolutional layer by RRCNN block. We used InstanceNorm3d with batch size 1 and applied RandomRotation, Mirroring for each iteration.

Table 5: Comparison of 4 models.

Filters	Contracting path	Expansive path	Upsampling pathway	DSC
10	5 RRCNNs	4 RRCNNs	-	0.580
16	5 RRCNNs	-	-	0.688
16	4 RRCNNs	-	0	0.716
16	5 RRCNNs (Replaced)	-	0	0.716

5. Discussion

Our validation results in the context of the HECKTOR challenge are summarized in Table 1 to Table 5. First, setting the suitable hyperparameters for the model is crucial. When dealing with 3D images, it would be better to use InstanceNorm3d rather than BatchNorm3d with batch size 1 in limited GPU usage. Secondly, adding and subtracting RRCNN block and Upsampling pathway to this

model possibly worsen the performance. Since we reduced the number of filters 24 to 16 or 10, this might be the reason. Hence, large number of filters extract more features leading to the better performance of the model. Recurrent model ensures better feature representation for segmentation tasks with few parameters. However, adding recurrent blocks or replacing the convolutional layers by recurrent blocks while reducing the number of filters is not desirable approach. RRSE U-Net showed the best performance with DSC score of 0.716, as shown in Table 5.

6. Conclusion

In this study, we developed deep CNNs to segment the tumor with the concatenated PET and CT images in the context of the MICCAI 2021 Head and Neck Tumor segmentation challenge (HECKTOR). We verified their feasibility using fully automated methods for medical image segmentation.

Given that the release date of the test cases is Aug. 01 2021 and the submission date is Sept. 01 2021, we used the training cases first to validate our models. For future work, we will test our model with the test set from AI crowd, MICCAI 2021 HECTOR challenge. A lot of U-Net based architectures have been proposed from all around the world. Such as V-Net, Scale Attention Network, Combining CNN and hybrid active contours, Two-stage approach for segmentation, GAN-based bi-modal segmentation, Patch-based 3D U-Net, Iteratively refine the segmentation. We will improve our model's performance by using these ideas.