Automatic Lesion segmentation in Whole-Body PET-CT using Coarse and Fine Segmentation models

Abdul Qayyum[1], Moona Mazher[2], Imran Razzak[3]

[1] ENIB, UMR CNRS 6285 LabSTICC, Brest, 29238, France
[2] Department of Computer Engineering and Mathematics, University Rovira I Virgili, Spain
[3]UNSW, Sydney, Australia

Abstract: Positron Emission Tomography / Computed Tomography (PET/CT) is an integral part of the diagnostic workup for various malignant solid tumor entities. Due to its wide applicability, Fluorodeoxyglucose (FDG) is the most widely used PET tracer in an oncological setting reflecting glucose consumption of tissues, e.g. typically increased glucose consumption of tumor lesions. To segment tumors from PET-CT in Whole-Body PET-CT, we have proposed a coarse and fine segmentation model. The 3DResUnet has been used for coarse segmentation and nnUNet model used for Fine segmentation. The prediction of the coarse segmentation model is fed into Fine segmentation to get the final prediction. The source code will be publicly available at

https://github.com/RespectKnowledge/Coarse-and-Fine-Segmentation_AutoPET

Proposed approach and Implementation Details

The main finding of the challenge is

1. Developed 3DResUNet with deep supervision for coarse segmentation
2. Use nnUNet for fine segmentation
3. Proposed 3DResUNet with deep supervision used for coarse segmentation and concatenated the output of coarse segmentation with nnUNet to get the fine segmentation output.
4. Two institutions dataset were used for training and validation of our proposed approach

The detailed description is shown in Figure.1.
The proposed solution consisted of two models, the first model in stage1 provided the coarse segmentation output and the second model used this coarse segmentation output coming from the model1.
3D-ResUnet with Deep Supervision: A framework of the proposed model is presented as an encoder, a decoder, and a baseline module. The 1x1 convolutional layer with softmax function has been used at the end of the proposed model. The 3D strides convolutional layer has been used to reduce the input image spatial size. The convolutional block consists of convolutional layers with Batch-Normalization and ReLU activation functions to extract the different feature maps from each block on the encoder side. In the encoder block, the spatial input size has been reduced with an increasing number of feature maps and on the decoder side, the input image spatial size will increase using a 3D Conv-Transpose layer. The input features' maps that are obtained from every encoder block are concatenated with every decoder block feature map to reconstruct the semantic information. The convolutional (3x3x3conv-BN-ReLu)

layer used the input feature maps extracted from every convolutional block on the encoder side and further passed these feature maps into the proposed residual module. The spatial size doubled at every decoder block and feature maps are halved at each decoder stage of the proposed model. The residual Bloch has been inserted at each encoder block with skip connection. The feature concatenation has been done at every encoder and decoder block except the last 1x1 convolutional layer. The three-level deep-supervision technique is applied to get the aggregate loss between ground truth and prediction. We have used nnUNet with cross-validation and selected the best fold for feta2022 segmentation, we have modified training and optimization parameters as compared to the original nnUNet. The batch size in nnUNet was 128x128x128 using 500 epochs.

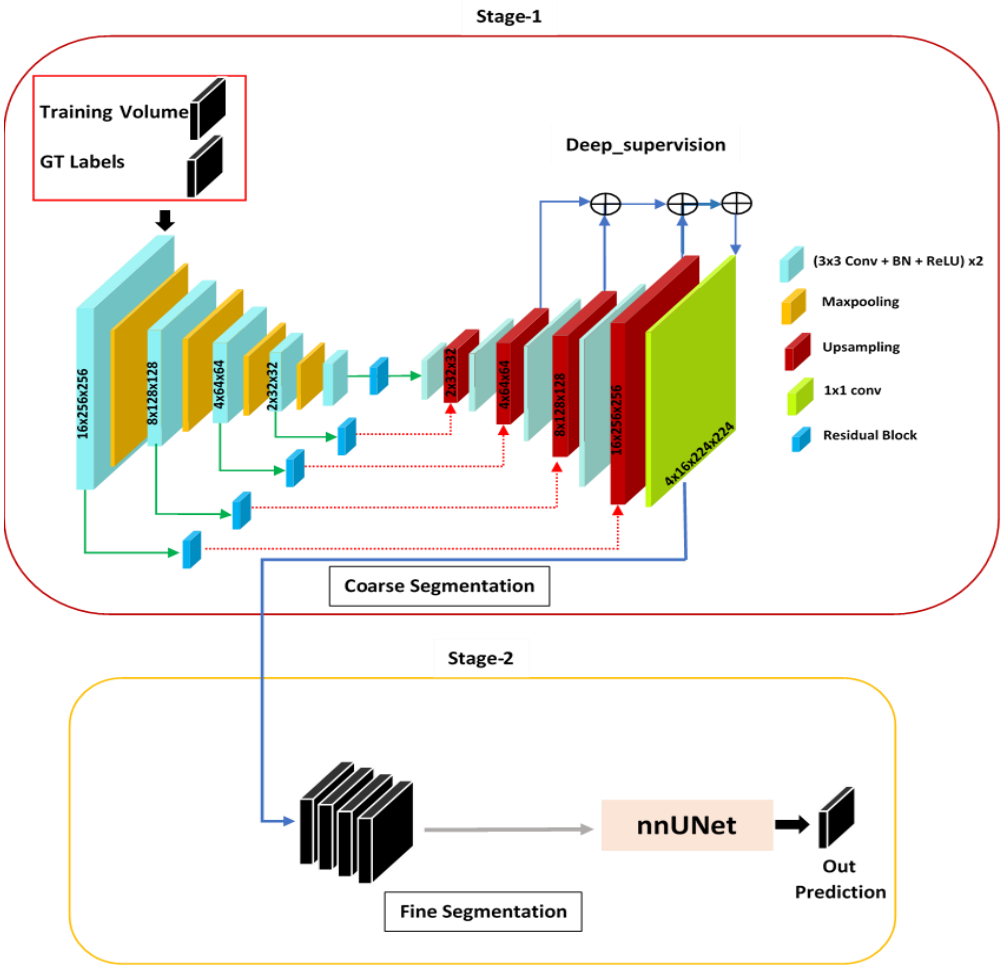

Figure.1. The proposed solution for the AutoPET2022 segmentation model.

Implementation Details

Environments and requirements

The proposed deep learning model is implemented in PyTorch and other libraries based on python are used for preprocessing and analysis of the datasets. The SimpleITK is used for reading and writing the nifty data volume. The ITK-SNAP is used for data visualization.

The environments and requirements of the proposed method are shown in Table 1.

Table 2. Environments and requirements.

CPU	Intel(R) Core (TM) i9-7900X CPU@3.30GHz
RAM	16×2GB
GPU	Nvidia V100
CUDA version	11
Programming language	Python3.7
Deep learning framework	Pytorch (Torch 1.7.0, torchvision 0.2.2)
Specification of dependencies	SimpleITK, Numpy, Skimage, Scipy, Nibabel, ITK-SNAP

Training protocols

The learning rate of 0.0004 with Adam optimizer has been for training the proposed coarse model. The binary cross-entropy function is used as a loss function between the output of the model and the ground-truth sample. 48 batch-size with 200 epochs has been used with 20 early stopping steps. The best model weights have been saved for prediction in the validation phase. The 128x128x128 input image size was used for training and prediction resample with the original input size at prediction using the nearest-neighbor interpolation method. The Pytorch library is used for model development, training, optimization, and testing. The V100 tesla NVidia-GPU machine is used for training and testing the proposed model. The data augmentation methods mentioned in Table.2. are used to further improve the results. The dataset cases have different intensity ranges. The dataset is normalized between 0 and 1 using the max and min intensity normalization method. The detail of the training protocol is shown in Table.2

Table 2. Training protocols.

Data augmentation methods	HorizontalFlip (p=0.5), VerticalFlip (p=0.5), RandomGamma (p=0.8)
Initialization of the network	“he” normal initialization

Patch sampling strategy	None
Batch size	2
Patch size	128x128x128
Total epochs	200
Optimizer	Adam
Initial learning rate	0.0004
Learning rate decay schedule	None
Stopping criteria, and optimal model selection criteria	The stopping criterion is reaching the maximum number of epochs (200).
Training time	10 hours

Testing protocols

The same preprocessing has been applied at testing time. The training size of each image is fixed (128x128x128) and used linear interpolation method to resample the prediction mask to the original shape for each validation volume. The prediction mask produced by our proposed model has been resampled such that it has the same size and spacing as the original image and copies all of the meta-data, i.e., origin, direction, orientation, etc.,