Joint low-count PET/CT segmentation and reconstruction with paired variational neural networks

Hongki Lim, Yuni K. Dewaraja, Jeffrey A. Fessler

University of Michigan

SPIE Medical Imaging 2020

Challenges of PET segmentation

- PET-based segmentation:
  Challenging due to inherent noise and poor spatial resolution.
- Manual segmentation on morphological (CT or MRI) images drawn by radiologist:
  1) time consuming 2) labor intensive
  3) high intra- and inter-observer variability

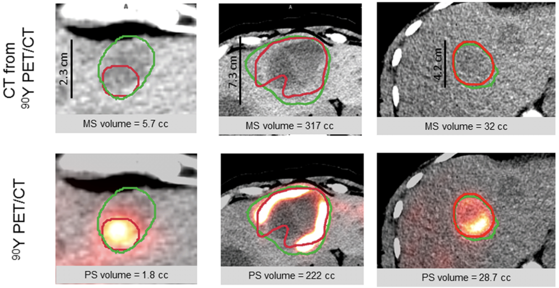


Fig. 2 Three examples (**a**–**c**) showing PET-based (red) and morphologic (green) segmentations on a single axial slice. Note the scale is different across the three cases and the metrics were evaluated over the full 3D extent of the VOIs

Figure: Lesion segmentation: A factor of significant variability in Y-90 radioembolization dosimetry[1]

[1] Mikell, Justin K., et al. "Impact of Y-90 PET gradient-based tumor segmentation on voxel-level dosimetry in liver radioembolization." EJNMMI physics 5.1 (2018): 31.

Problem statement

- Limitation of existing fully automatic medical imaging segmentation:
  - Perform after reconstruction.

    : Propagation of errors from noisy reconstructions to the segmentation step
  - Use a single modality.

    : Not fully exploit the information from dual-modality systems like PET/CT.

$\rightarrow$ Fully automatic joint segmentation-reconstruction using multi-modality images is desirable

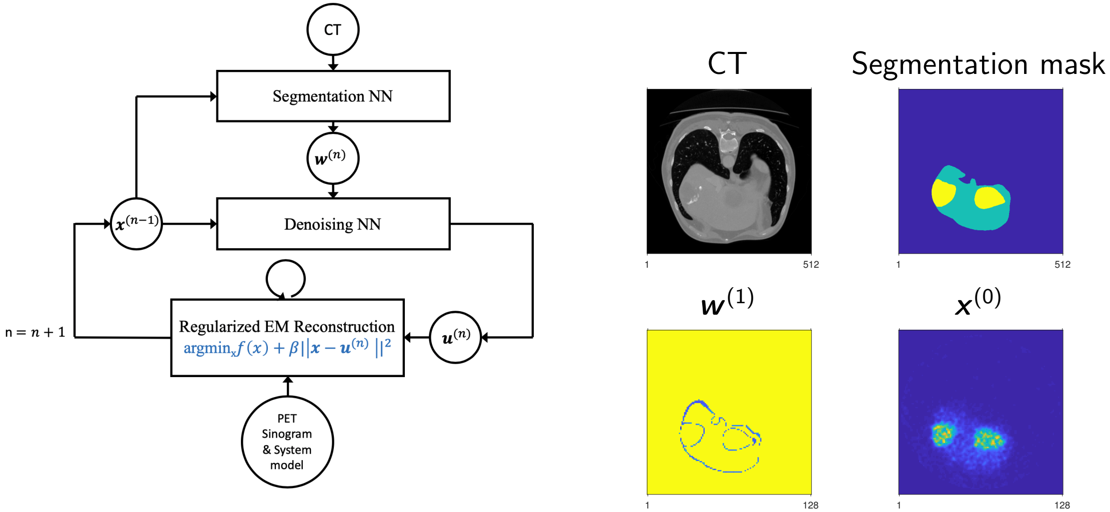


Figure: High-level block diagram of proposed method

Joint multimodal segmentation and reconstruction framework

- The framework is inspired by following optimization problem:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \beta \mathrm{R}(\boldsymbol{x}; \boldsymbol{w})$$

  - $\boldsymbol{x}$: unknown PET image
  - $f(\boldsymbol{x})$: data fidelity term
    $\rightarrow$ Poisson negative log-likelihood for measurement $\boldsymbol{y}$
    & estimated measurement means $\bar{\boldsymbol{y}}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \bar{\boldsymbol{r}}$
  - $\boldsymbol{A}$: system model
  - $\bar{\boldsymbol{r}}$: mean background events
  - β: regularization parameter
  - $\mathrm{R}(\boldsymbol{x}; \boldsymbol{w})$: regularization term

Regularization

- The regularization term is composed of convolutional operations followed by a thresholding operation to promote sparsity:

$$\mathrm{R}(\boldsymbol{x};\boldsymbol{w}) = \sum_{k=1}^{K} \frac{1}{2} \left\| \boldsymbol{c}_k * \boldsymbol{x} - \boldsymbol{z}_k \right\|_{\boldsymbol{W}}^2 + \alpha_k \left\| \boldsymbol{z}_k \right\|_1 .$$

  - $\boldsymbol{w}$ is a boundary indicator image (zero-valued at boundary) given by segmentation network utilizing both CT and PET modalities
  - $\boldsymbol{W} = \mathrm{diag}\{\boldsymbol{w}\}$

Idea behind the formulation

- Two equivalent formulations:

$$\hat{x} = \arg\min_{\boldsymbol{x}} \min_{\boldsymbol{z}} f(\boldsymbol{x}) + \beta\Big(\sum_{k=1}^{K} \frac{1}{2}\|\boldsymbol{c}_k * \boldsymbol{x} - \boldsymbol{z}_k\|_{\boldsymbol{w}}^2 + \alpha_k\|\boldsymbol{z}_k\|_1\Big)$$

$$\Updownarrow$$

$$\hat{x} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \beta\sum_{k=1}^{K} \mathcal{H}(\boldsymbol{c}_k * \boldsymbol{x},\ \alpha_k,\ \boldsymbol{w})$$

- $\mathcal{H}(\cdot,\cdot,\cdot)$ is the Huber function:

$$\mathcal{H}(\boldsymbol{t}, p, \boldsymbol{q}) = \sum_j q_j h(t_j, \frac{p}{q_j}), \quad h(t,\delta) = \begin{cases} \frac{t^2}{2}, & |t| < \delta \\ \delta|t| - \frac{\delta^2}{2}, & |t| \geq \delta. \end{cases}$$

$\rightarrow$ Not penalize the filtered image at jth voxel when $w_j = 0$ (edge area).

Variable updates

- Block coordinate descent algorithm alternatively updates $\{\boldsymbol{z}_k : \boldsymbol{z}_1, \ldots, \boldsymbol{z}_K\}$ and $\boldsymbol{x}$:

$$\boldsymbol{z}_k^{(n+1)} = \arg\min_{\boldsymbol{z}_k} \frac{1}{2} \left\| \boldsymbol{c}_k * \boldsymbol{x}^{(n)} - \boldsymbol{z}_k \right\|_{\boldsymbol{W}}^2 + \alpha_k \|\boldsymbol{z}_k\|_1$$

$$= \mathcal{T}\left(\boldsymbol{c}_k * \boldsymbol{x}^{(n)}, \alpha_k \oslash \boldsymbol{w}\right)$$

$$\boldsymbol{x}^{(n+1)} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{\beta}{2} \left(\sum_{k=1}^{K} \left\| \boldsymbol{c}_k * \boldsymbol{x} - \boldsymbol{z}_k^{(n+1)} \right\|_{\boldsymbol{W}}^2 \right),$$

  - $\mathcal{T}(\cdot, \cdot)$ is the element-wise soft thresholding operator:

$$\mathcal{T}(\boldsymbol{t}, \boldsymbol{q})_j := \mathrm{sign}(t_j) \max(|t_j| - q_j, 0).$$

$\rightarrow$ $\boldsymbol{W}$ is designed to avoid smoothing across boundaries between different regions.

Equivalent form of updates under some conditions

- Two equivalent updates:

$$\boldsymbol{z}_k^{(n+1)} = \mathcal{T}(\boldsymbol{c}_k * \boldsymbol{x}^{(n)}, \alpha_k \oslash \boldsymbol{w})$$

$$\boldsymbol{x}^{(n+1)} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{\beta}{2}\left(\sum_{k=1}^{K} \|\boldsymbol{c}_k * \boldsymbol{x} - \boldsymbol{z}_k^{(n+1)}\|_{\boldsymbol{W}}^2\right)$$

$$\Updownarrow \sum_{k=1}^{K} \boldsymbol{C}_k^T \boldsymbol{W} \boldsymbol{C}_k = \boldsymbol{I} \quad (\boldsymbol{C}_k \boldsymbol{x} \iff \boldsymbol{c}_k * \boldsymbol{x})$$

$$\boldsymbol{u}^{(n+1)} = \sum_{k=1}^{K} \tilde{\boldsymbol{c}}_k * \left(\boldsymbol{W}\left(\mathcal{T}(\boldsymbol{c}_k * \boldsymbol{x}^{(n)}, \alpha_k \oslash \boldsymbol{w})\right)\right)$$

$$\boldsymbol{x}^{(n+1)} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{\beta}{2}\|\boldsymbol{x} - \boldsymbol{u}^{(n+1)}\|_2^2,$$

where $\tilde{\boldsymbol{c}}_k^{(n)}$ is flipped version of $\boldsymbol{c}_k^{(n)}$

Training denoising network

- Train the set of filters $\{\boldsymbol{c}_k\}$, $\{\boldsymbol{d}_k\}$ and soft-thresholding values $\{\alpha_k\}$ to map the previously estimated image to high quality image at each iteration:

$$\{\hat{\boldsymbol{c}}_k{}^{(n+1)}\}, \{\hat{\boldsymbol{d}}_k{}^{(n+1)}\}, \{\hat{\alpha}_k{}^{(n+1)}\} = \underset{\{\boldsymbol{c}_k\},\{\boldsymbol{d}_k\},\{\alpha_k\}}{\arg\min} \left\| \boldsymbol{x}_{\text{true}} - \sum_{k=1}^{K} \boldsymbol{d}_k * \left(\boldsymbol{W} \left(\mathcal{T} (\boldsymbol{c}_k * \boldsymbol{x}^{(n)}, \alpha_k \oslash \boldsymbol{w}) \right) \right) \right\|_2^2 .$$

- Filter size: $3 \times 3 \times 3$.
- Number of filters at each iteration: $K = 192$.
- Trained using Adam optimization with PyTorch deep learning library.

Details on segmentation network

- We implemented 3-D version of U-Net.
- Input of segmentation network is four dimensional array:
  channel $\times$ image depth (16) $\times$ image height (512) $\times$ image width (512)
  - 2 channels: [CT image; PET image upsampled to CT size]
  - Training loss: Cross-entropy

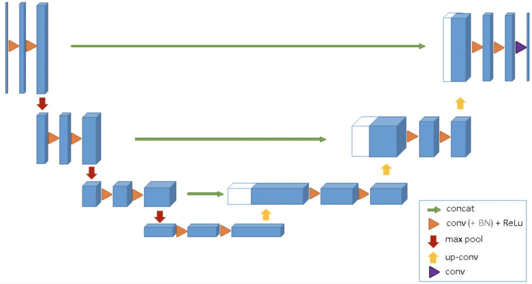


Figure: 3D U-Net architecture

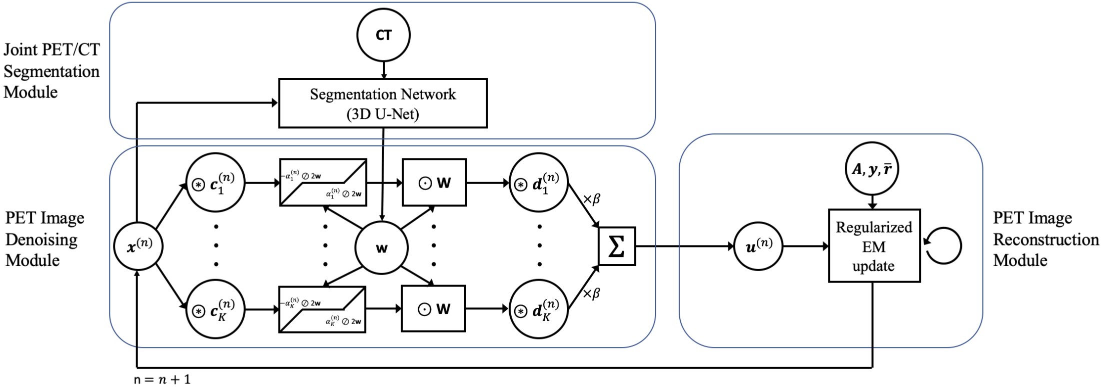


Figure: Detailed block diagram of proposed method. Final output image is from the reconstruction module.

Dataset

- We used the publicly opened dataset LiTS announced in the MICCAI 2017.
- We use 15 samples for training and 4 samples for testing.

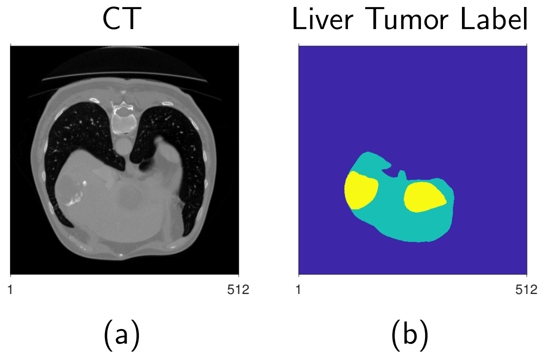
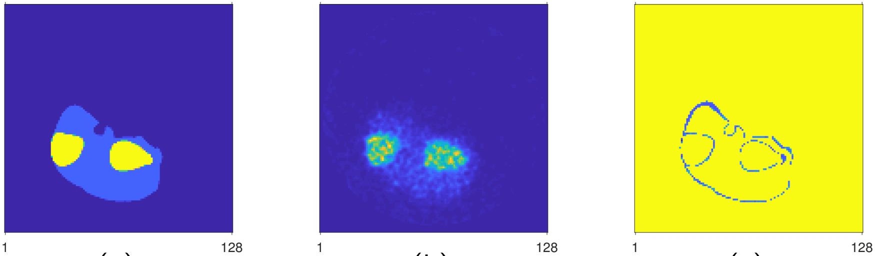


Figure: LiTS dataset provides CT image (a) in HU unit and label image (b) (tumor:2, liver:1) corresponding to CT.

Generating PET data

- Used the ground-truth label images to generate the true activity maps (changed the tumor value to 5) and the corresponding synthetic PET data.

PET true image | PET reconstructed image using EM ($\boldsymbol{x}^{(0)}$) | True boundary $\boldsymbol{w}$



(a) (b) (c)

Figure: We generated PET image (b) simulating Y-90 PET after radioembolization. We set the tumor-to-liver ratio as 5:1. Image (c) shows true boundary based on label image. Boundary is zero-valued.

Evaluation metrics

- PET reconstruction evaluation:

$$\mathrm{CNR} = \frac{C_{\mathrm{Lesion}} - C_{\mathrm{Liver}}}{STD_{\mathrm{Liver}}}$$

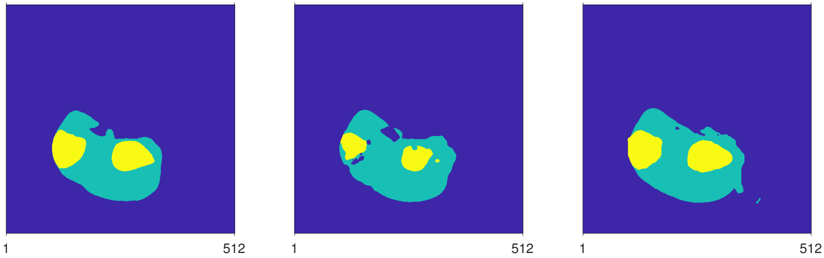
$$\mathrm{RMSE}\ (\%) = \sqrt{\frac{\sum_j (x_{\mathrm{true}}[j] - \hat{x}[j])^2}{J_{\mathrm{FOV}}}} \times 100,$$

- Segmentation evaluation:

$$\mathrm{Dice}(A, B) = \frac{2|A \cap B|}{|A| + |B|},$$

Segmentation result (4 test cases averaged)

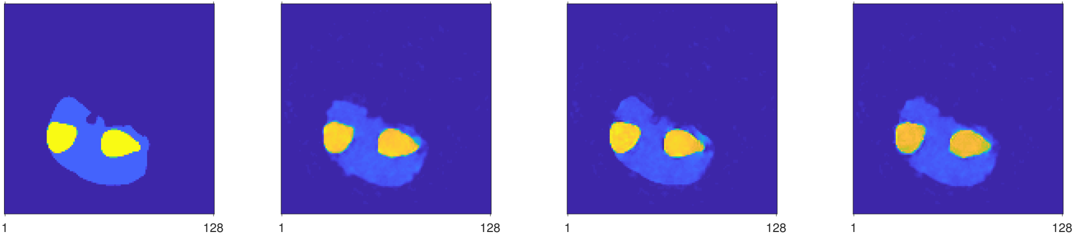
True Segmentation | Segmentation w/ single-modality (CT) | Segmentation w/ multi-modality (PET&CT)



Method	Dice tumor	Dice liver
Proposed method using single-modality (CT)	0.51	0.92
Proposed method using multi-modality (PET/CT)	**0.87**	**0.93**

Reconstruction result (4 test cases averaged)

True PET | PET reconstruction w/o $\boldsymbol{w}$ | PET reconstruction w/ true $\boldsymbol{w}$ | PET reconstruction w/ estimated $\boldsymbol{w}$



Method	CNR	RMSE
EM	8.72	14.36
Proposed method w/o boundary	9.62	**8.41**
Proposed method w/ estimated boundary	**9.89**	**8.41**
Proposed method w/ true boundary	11.2	7.30

More to investigate

- Impact of misregistration between modalities
  - Liver is located nearby lung, therefore respiratory motion may bring misregistration problem especially in upper part of liver
- Inconsistency in activity distribution
  - Tumor-to-liver ratio
  - Non-uniformity in tumor and background liver

Future works

- Training neural networks for segmentation and denoising together with a weighted combination of loss functions.
- Use shared weights (or representation) between the segmentation and denoising networks.[1]

[1] Lee, S., Stokes, J., Eaton, E. (2019, August). Learning shared knowledge for deep lifelong learning using deconvolutional networks. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (pp. 2837-2844). AAAI Press.

Thank You