**SupplementaL Material**

**table of contents**

**Supplementary Methods**

**Supplementary Table 1.** List of institutions from which the MR images were acquired utilized in this study with demographic information.

**Supplementary Table 2.** The parameters of 8 image processing techniques utilized for the data augmentation. The notation *U* denotes a uniform distribution from which the parameters were randomly chosen.

**Supplementary Table 3.** Python library modules (software environment) and their versions for implementing the deep learning model.

**Supplementary Figure 1.** Schematic diagram of the U-Net like deep learning architecture utilized in this study. The feature map size and the target voxel spacing (mm) are denoted on the right of each convolutional block, and each convolutional block consists of convolution, instance normalization, and activation function (Leaky ReLU). Two consecutive convolutional blocks are used per resolution stage, and the target (median) voxel size and spacing were 378×395×46 and 4×0.89×0.89, respectively.

**Supplementary METHODS**

**Dataset Distribution**

The HALT Polycystic Kidney Disease (HALT-PKD) study is a multi-institutional clinical trial, and the MR images utilized for network training and performance evaluation were acquired from 8 different sites. Supplementary Table 1 shows a list of institutions and demographic information of the dataset from each institution.

**Data Augmentation**

The data augmentation was performed with 8 image processing techniques: 3 spatial transforms (rotation, scaling, and flip); 3 intensity transforms (brightness, contrast, and Gamma correction); and 2 image degradations (additive noise and blurring). The spatial transforms were applied along all 3D axes, and for intensity transforms and image degradations, the voxel intensity values were normalized to have ranges between 0 and 1 prior to the data augmentation. The parameters of all transforms were randomly chosen from the predefined probabilities and intervals as listed in Supplementary Table 2.

**Deep Learning Architecture and Parameters**

Supplementary Figure 1 shows the U-Net like deep learning architecture utilized in this study. Each convolutional block consists of 3D convolution, 3D instance normalization, and activation function (Leaky ReLU) where two consecutive convolutional blocks are deployed per resolution stage in both contraction and expansion layers. The downsampling and upsampling for contraction and expansion layers are done with the strided convolution (stride=2) and convolution transposed, respectively, and the output of each convolutional block are concatenated with the corresponding resolution stage as the skip connection. The target (median) voxel size and spacing were 378×395×46 and 4×0.89×0.89, respectively, for which the data were resampled using third order spline interpolation. The output layer using 1×1×1 convolution predicts the voxel to be one of four categories (i.e., background, right kidney, left kidney, and exophytic cyst).

 For the training of the deep learning model, stochastic gradient descent was utilized as the optimizer with a Nesterov momentum of 0.99 and learning rate of 0.01, and the learning rate is decreased to nearly 0 during the training according to the poly learning policy.1 The maximum number of epoch was fixed to 1,000 with 250 training iterations, and the combination of the Dice loss and a cross-entropy loss was used for the evaluation. The deep learning architecture was implemented in the hardware environment with 4 Nvidia TitanXp GPUs, 128 GB RAM, and Intel Xeon Silver 4110 CPU. In addition, the Python library modules and their versions exploited in this study were listed in Supplementary Table 3.

**REFERENCES**

1. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40: 834–848, 2018

2. batchgenerators by MIC@DKFZ [Internet]. MIC-DKFZ; Available from: https://github.com/MIC-DKFZ/batchgenerators

Supplementary Table 1. List of institutions from which the MR images were acquired utilized in this study with demographic information.

|  |  |  |  |
| --- | --- | --- | --- |
| Institution | Mean Age±SD | Sex | Total# |
| Male | Female |
| Beth Israel Deaconess Medical Center | 36.9±5.8 | 3 | 9 | 12 |
| Cleveland Clinic | 37.2±6.4 | 6 | 12 | 18 |
| University of Colorado Hospital | 38.1±8.2 | 9 | 11 | 20 |
| Emory University Hospital | 38.3±6.8 | 15 | 29 | 44 |
| Kansas Medical Center | 42.5±5.5 | 27 | 12 | 39 |
| Mayo Clinic | 39.7±5.1 | 12 | 15 | 27 |
| Tufts Medical Center | 40.8±6.7 | 24 | 18 | 42 |
| University of Alabama Hospital | 35.7±5.7 | 3 | 5 | 8 |

Supplementary Table 2. The parameters of 8 image processing techniques utilized for the data augmentation. The notation *U* denotes a uniform distribution from which the parameters were randomly chosen.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Technique | Probability | Range |
| Spatial Transform | Rotation | 0.2 | *U*[-25, 25] |
| Scaling | 0.2 | *U*[0.5, 1.5] |
| Flip | 0.5 | ­­­– |
| Intensity Transform | Brightness | 0.25 | *U*[0.6, 1.4] |
| Contrast | 0.25 | *U*[0.7, 1.3] |
| Gamma Correction | 0.2 | *U*[0.7, 1.3] |
| Image Degradation | Additive Noise | 0.2 | *U*[0, 0.15] |
| Blurring | 0.2 | *U*[0.5, 2] |

Supplementary Table 3. Python library modules (software environment) and their versions for implementing the deep learning model.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version | Name | Version |
| OS (Ubuntu) | 16.04 LTS | MedPy | 0.4.0 |
| Python | 3.8.5 | batchgenerators2 | 0.21 |
| PyTorch | 1.7.1 | NumPy | 1.19.2 |
| tqdm | 4.57.0 | scikit-learn | 0.24.1 |
| dicom2nifti | 2.2.12 | SimpleITK | 2.0.2 |
| scikit-image | 0.18.1 | pandas | 1.2.2 |
| SciPy | 0.24.1 | NiBabel | 3.2.1 |

|  |
| --- |
|  |
| Supplementary Figure 1. Schematic diagram of the U-Net like deep learning architecture utilized in this study. The feature map size and the target voxel spacing (mm) are denoted on the right of each convolutional block, and each convolutional block consists of convolution, instance normalization, and activation function (Leaky ReLU). Two consecutive convolutional blocks are used per resolution stage, and the target (median) voxel size and spacing were 378×395×46 and 4×0.89×0.89, respectively.  |
|  |