SUPPLEMENTARY MATERIALS 3

Deep Learning with YOLOv3

YOLO is a deep convolutional neural network (CNN)-based image recognition model for object detection and classification. In this study, we used the most recently updated version of YOLO (https://github.com/pjreddie/darknet). The previous versions, YOLOv1 and YOLOv2, are well-known for their overwhelming real-time detection speed and satisfactory accuracy [3,4]. YOLOv3 uses Darknet-53, a 53-layer network trained on ImageNet, as the backbone, and eventually possesses an architecture comprising 106 fully convolutional layers [5]. It also uses residual blocks and skip connections, absent in previous versions. However, this complex YOLOv3 architecture caused some loss in object detection speed, inducing a marked improvement in accuracy [6,7]. Another advantage of YOLOv3 is that it detects objects on three different scales. By using these multiscale detection methods, YOLOv3 can overcome the weaknesses of the previous versions in the detection of small objects [5].

Hardware Environment for YOLOv3

The deep learning-based lesion detection algorithm (DLLD) including YOLOv3 was implemented in the following hardware environment: PowerEdge T630 (Dell) with four Titan-V graphic cards (NVIDIA) and software environment: Linux Ubuntu 16.04.4 LTS, NVIDIA driver version 396.26, CUDA version 9.1.

CT Image Conversion for YOLOv3

CT images of the patients were converted to the PNG image format at the window setting (level, 50 Hounsfield unit [HU]; width, 250 HU) optimized to abdominal soft tissues for use as the DLLD input. Total CT images of the validation group were used as input images during the validation process.

Training Parameters of YOLOv3

YOLOv3 training can be configured using the file named "yolov3-voc.cfg. The main configuration details are as follows:

```
[net]
# Training
batch = 64
subdivisions = 16
width = 416
height = 416
channels = 3
momentum = 0.9
decay = 0.0005
angle = 0
saturation = 1.5
exposure = 1.5
hue = .1
learning_rate = 0.00025 # for 4 GPUs
burn_in = 4000 # for 4 GPUs
max_batches = 50200
policy = steps
steps = 40000,45000
scales = .1,.1
```

YOLOv3 was trained for approximately 950 epochs using a training dataset. Supplementary Figure 1 depicts the training loss curve of the proposed YOLOv3. After the early fast-learning period, the loss continuously and gradually decreased. After

the 800th epoch, it almost converged. The final trained-weight volume of YOLOv3 was selected at the 847th epoch, one of the points with the lowest loss values after the 800th epoch.

Random sampling and random augmentation were performed for overcoming the imbalance in the number of input images for each class, to make the three classes (metastasis, cyst, and hemangioma) have 5000 input images each [8,9]. During the training process, random runtime augmentation was performed to improve the generalizability.

C++ language was used to run the YOLOv3 model, and the other preprocessing was implemented on Python (version 3.6).