**Supplementary Materials**

**Lymph node detection model (LNDM)** **based on faster R-CNN**

Deep convolutional neural networks (CNNs) have achieved success in various computer vision tasks like object recognition, motion analysis, and scene reconstruction. The task of detection of abnormal lymph nodes in CT belongs to object detection. Region-based methods of detection that use CNN are now the state of the art in object detection. The latest generation of region-based detection methods, such as the faster R-CNN, has delivered impressive results on various object detection benchmarks. By training a faster R-CNN on our abnormal lymph node dataset, we constructed an LNDM based on faster R-CNN. Specifically, the LNDM consists of two modules: (1) regional proposal network (RPN), which is a fully convolutional network for generating object proposals that are fed into the second module; and (2) the faster R-CNN detector, which detects objects from these proposals. RPN and faster R-CNN share convolutional layers that generate feature maps [Supplementary Figure 1].

The RPN consists of two fully connected sibling layers for the regression and classification of the proposal bounding box. The input is a 3 × 3 sliding window in the last layer of the feature map. Two 1 × 1 convolutional layers are then added for the classification and regression branches of all sliding windows. To deal with different scales and aspect ratios of objects, anchors are introduced to the RPN. An anchor is at the center of each sliding window on the feature maps. Each anchor has a specific scale (8, 16, and 32) and an aspect ratio (1:1, 1:2, and 2:1). There are nine anchors in total at each location in the design. Therefore, for a convolutional feature map of size W × H, we have at most W × H × 9 possible proposals. The same features of each sliding window are used to regress nine proposals, instead of extracting multiple sets of features and training a single regressor. The RPN can be trained in an end-to-end manner using back-propagation and stochastic gradient descent (SGD). The faster R-CNN detector shares convolutional layers with the RPN.

The objective function is $L\left(\left\{p\_{i}\right\},\left\{t\_{i}\right\}\right)=\frac{1}{N\_{cls}}\sum\_{i}^{}L\_{cls}\left(p\_{i},p\_{i}^{∗}\right)+λ\frac{1}{N\_{reg}}\sum\_{i}^{}p\_{i}^{∗}L\_{reg}(t\_{i},t\_{i}^{∗})$, where $p\_{i}$is the predicted probability of the *i*-th anchor in a mini-batch being an object, and $p\_{i}^{∗}$is the ground truth of the anchor. $t\_{i}$ consists of four numbers $(t\_{x},t\_{y},t\_{w},t\_{ℎ})$ that represent the coordinates and size of the *i*-th predicted bounding box, and $t\_{i}^{∗}$ is the ground truth of a bounding box with a positive anchor. $L\_{cls}$ is the classification log loss over two classes and $L\_{reg}$ is the smooth L1 loss function for regression. These two losses are normalized by $N\_{cls}$ (mini-batch size) and $N\_{reg}$ (number of anchor locations) with weight parameter $λ$. This means that the classification and regression terms roughly have the same weight. $p\_{i}^{∗}L\_{reg}$ guarantees that regression loss is activated only for positive anchors.

**Training process of lymph node detection model**

In this paper, we use the RPN, an approximate four-step end-to-end joint training strategy proposed in using of SGD.

Step 1. We initialize the network with a VGG16 model pre-trained on ImageNet and fine-tune it end to end on the region proposal task using CT images and annotations of abnormal lymph nodes.

Step 2. We train a detection network of faster R-CNN using the proposals generated in step 1. This detection network is also initialized by pretrained weights of the VGG16 model. The detection network does not share convolutional layers with the RPN.

Step 3. Networks were used to initialize Regional Proposal Network training. The shared convolutional layers were fixed and the Regional Proposal Network layers were fine-tuned. In this step, the Regional Proposal Network and detection networks share convolutional layers.

Step 4. Keeping the shared convolutional layers fixed, the detection network was fine-tuned. Thus, both networks shared the same convolutional layers and were trained as a unified network.

We trained our model using the training dataset following the steps described above.

**Experimental Settings**

The experiment was run on a server equipped with an Intel Xeon CPU E5-2697 with 2.60 GHz and an NVIDIA Tesla P100 GPU with 16 GB of memory. The graphics card provided powerful acceleration effects for data calculations in the experiment.

**Supplementary Table 1: Training parameters of faster R-CNN.**

|  |  |
| --- | --- |
| **Variables** | **Values** |
| Iteration | 50,000  |
| Learning rate | 0.001 before 30 K iterations |
|  | 0.0001 after 30 K to 50 K iterations |
| Momentum | 0.9 |
| Weight decay | 0.0005 |
| Scale of anchor | 8, 16, 32 |
| Aspect ratio of anchor | 1:1, 2:1 |

**Supplementary Table 2: The parameters results of model in subgroup study of lymph node size and location.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Characteristics** | **Validation dataset (*n* = 3260)** | **Sensitivity (95% CI)** | **Specificity (95% CI)** | **AUC (95% CI)** |
| LN diameter (mm) |  |  |  |  |
| 0–10 | 2029 (62.2) | 0.853 (0.721–0.972) | 0.821 (0.768–0.892) | 0.905 (0.889–0.922) |
| >10 | 1231 (37.8) | 0.729 (0.620–0.878) | 0.983 (0.962–0.994) | 0.901 (0.881–0.922) |
| LN station, % (abnormal/total)\* |
| 2 | 59.9 (175/292) | 0.859 (0.811–0.920) | 0.821 (0.757–0.899) | 0.882 (0.838–0.926) |
| 3A | 29.9 (32/107) | 0.861 (0.811–0.921) | 0.797 (0.724–0.868) | 0.883 (0.819–0.946) |
| 3P | 29.3 (24/82) | 0.837 (0.776–0.899) | 0.665 (0.610–0.722) | 0.751 (0.626–0.877) |
| 4R | 46.2 (42/91) | 0.854 (0.806–0.919) | 0.865 (0.799–0.913) | 0.905 (0.872–0.939) |
| 4L | 58.4 (348/596) | 0.791 (0.722–0.885) | 0.800 (0.730–0.882) | 0.866 (0.813–0.918) |
| 5 | 42.4 (230/543) | 0.924 (0.859–0.996) | 0.813 (0.737–0.895) | 0.919 (0.894–0.944) |
| 6 | 26.4 (86/326) | 0.889 (0.823–0.965) | 0.912 (0.843–0.980) | 0.917 (0.878–0.955) |
| 7 | 48.8 (465/953) | 0.920 (0.840–0.989) | 0.969 (0.922–1.000) | 0.949 (0.934–0.965) |
| 8 | 54.6 (59/108) | 0.825 (0.736–0.908) | 0.773 (0.713–0.858) | 0.816 (0.735–0.897) |
| 9 | 63.0 (102/162) | 0.744 (0.676–0.812) | 0.930 (0.861–0.994) | 0.861 (0.803–0.921) |

\*Abnormal LN was determined by postoperative pathological examination. AUC: Area under the curve; CI: Confidence interval; LN: Lymph node.

**Supplementary Table 3: Comparison between LNDM and radiologists on diagnose of lymph node metastasis in lung cancer patients.**

|  |  |  |
| --- | --- | --- |
| **Characteristics** | **LNDM** | **Radiologists** |
| Time needed/patient (s) | 20–40 | 300–800 |
| Accuracy of normal LN (%) | 87.8 | 85.4 |
| Accuracy of abnormal LN (%) | 88.9 | 88.9 |
| AUC of ROC curve (95% CI) | 0.823 (0.683–0.964) | 0.812 (0.670–0.954) |
| Sensitivity (95% CI) | 0.908 (0.844–0.965) | 0.891 (0.764–0.957) |
| Specificity (95% CI) | 0.811 (0.766–0.903) | 0.798 (0.665–0.933) |

AUC: Area under the curve; CI: Confidence interval; LN: Lymph node; LNDM: Lymph node detection model; ROC: Receiver operating characteristic.



**Supplementary Figure 1:** The working characteristics of the faster region-based convolutional neural network. RPN: Regional proposal network; R-CNN: Region-based convolutional neural network.



**Supplementary Figure 2:** Flowchart of the training and validation datasets (A); Flowchart of the assessment cohort (B). LN: Lymph node; LNDM: Lymph node detection model.



**Supplementary Figure 3**: Training loss function values for the model.