DLCV Fall 2021 Final Challenge No.2: Nodule Detection

[1]Yen-Chih Huang [2]Ken Yu [3]Chiao Lai

[1]D06522014 [2]R10942152 [3]R09921005

Abstract

Our goal for this project is to detect nodules from CT scans which represented as a 3D array of single-channel data. The only information we have is the location (x, y, z) and diameter of each nodule in the training dataset. We studied two kinds of methods. One is a two stage method which contains a 2D-based detection function and a 3D-based classification model, and the second one is an end-to-end 3D-based object detection model. Finally, we use the two-stage method for solving this problem. The 2D detection part is forced to consume the entire CT scan to find all suspected nodules. Then, the 3D-based classification model gets a zoomed-in view of the areas of interest to classify the slicing voxels are either a real nodule or not.

2D Image Segmentation by CV

Preprocessing CT Image

Since nodules only take a very little proportion of original CT images, we need to segment out our region of interest first. We use conventional image processing techniques to help us to get regions of the lung in CT images. First, we use -400HU as threshold to binarize the image, and then clean the background by delete regions that touch the image border. Secondly, we extract two largest connected regions and assume it's the regions of the lung. Finally , we perform a closing operation to fill the little holes inside the mask and use that mask to get lung segments from the original image.

Use Conventional CV Technique to Find Nodule

After segmenting lung segments, we try to find suspicious nodules in the lung areas. However, we find it's very hard to tell whether it's a nodule or trachea by just viewing a single CT slice. Therefore, we superpose three neighbor CT slices together, making the trachea's sharp shape and nodules circular shape revealed, allowing us to filter out the trachea by some easy morphological operations. The morphological operations are as follows. We delete areas smaller than 10 pixels, and areas bigger than 800 pixels. And then, check if the shape is circular enough by its eccentricity. These simple operations are suffice to filter most of the trachea, and leave us with some suspicious nodules. The segmentation workflow of the CV method is shown as figure 1.

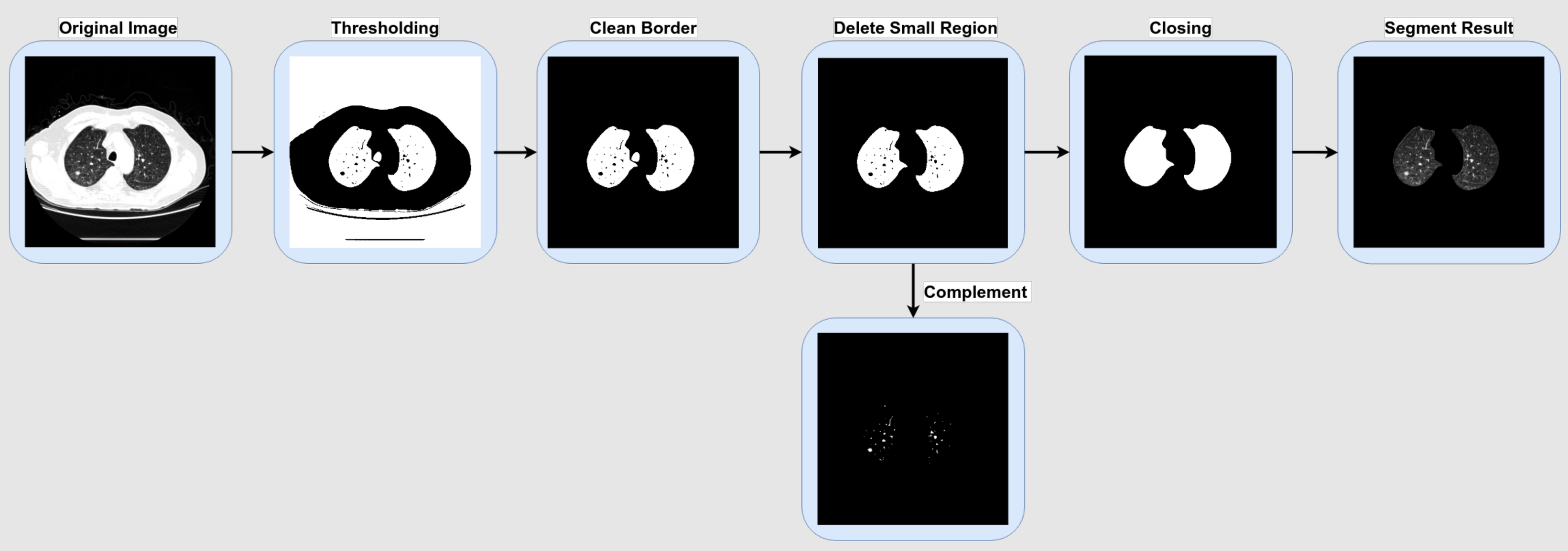

Figure 1. Segmentation workflow by CV

2D Image Segmentation by U-Net

Our anohter segmentation method is U-Net [Ronneberger et al. 2015] as shown in figure 2. The model is a design for a neural network that can produce pixel-wise output and that was invented for segmentation. By creating the 2D slice image from CT scan and masking the pixels which are annotated as nodules, we can use U-Net to perform nodule detection.

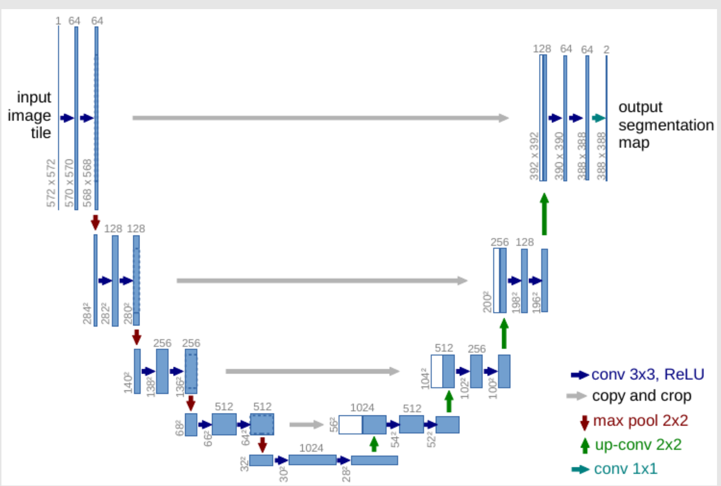

Figure 2. U-Net architecture [Ronneberger et al. 2015]

We setup the padding flag of the U-Net constructor to True. This will mean we can use input images of any size, and we will get output of the same size. But we may lose some fidelity near the edges of the image. Another problem is the model only accept a 3-channel 2D image input. Therefore, we can only treat each slice of CT scan as a 2D segmentation problem. Instead of the RGB information, our channels will be the slice we are actually segmenting, and the neighboring slice on each side.

Because U-Net can take images of arbitrary size, we can get away with training and validating on samples with different dimensions. Since each nodule is relative small compared to the whole CT slice. In order to aviod the class-balancing issue. We implement our training dataset by only taking randomly cropped images nearby the nodules. By training on crops, we can keep the number of positive pixels and negative pixels relatively the same.

Figure 3. Segmentation results of U-Net

Each pixel of U-Net's output is the model's estimate of the probability that the pixel in question is part of a nodule. We use a threshold of 0.5 to convert it into a binary image representing the positive prediction pixel. The result is shown as figure 3. We use red color for all pixels that are incorrect (false positives and false negatives). Every correctly predicted pixel inside a nodule is set to green (true positives).

3D-Based Nodule Classification

With the 2D segmentation results, we can use a connected-components algorithm to group the pixels into suspected nodules. Our CV segmentation method produce 133k suspected nodules from the training dataset, and U-Net is 80k. Both of those methods can only find as many nodules as possible. We need a classification model to tell a suspected nodule is a real nodule or not. The model [Stevens et al. 2020] architecture is shown as figure 4.

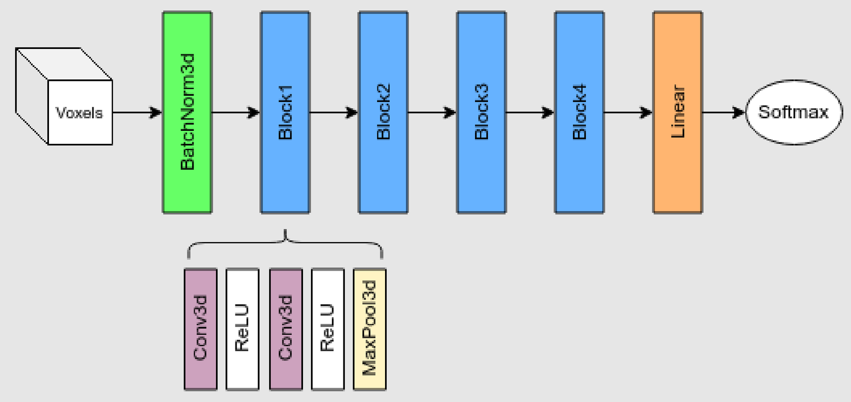

Figure 4. Classification model 3D-ConvNet

Based on the nodule's location, we can slice a 32x48x48 voxels from CT scan as the model's input. If the selecting nodule is not appear in real nodules (annotations.csv), then the training label will set to 0. The backbone of the network contains four repeated blocks. Each block has the same set of layers that consists of two 3x3x3 convolution layers, each followed by a ReLU activation, with a max-pooling operation at the end of the block. Finally, we use fully connected layer to convert the output of the backbone into a Softmax function to complete a single-label classification task.

Results

Detection	Classification	Pub.FROC	Pri.FROC
CV	-	0.014	0.012
U-Net	-	0.002	0.001
CV	3D-ConvNet	0.456	0.475
U-Net	3D-ConvNet		

Table 1. Final results of the project

3D-Based Object Detection

It's intuitive to use a 3D-based object detection method to find the nodule. The main idea of the 3D detection is similar to the 2-stage method in 2D detection. The most different part is to replace the 2D CNN modules with 3D CNN modules.

With the architecture similar to U-Net, the feature backbone extracts the 3D image features from the 3D CT images. After extracting the features, there is a 3D RPN (1. Nodule candidate screening) to find the ROIs and output the predicted bounding box ((x, y, z, d) in our case) and the predicted probability. Furthermore, to improve the performance for the model. There is a binary classification model to classify whether the ROIs are real nodule.

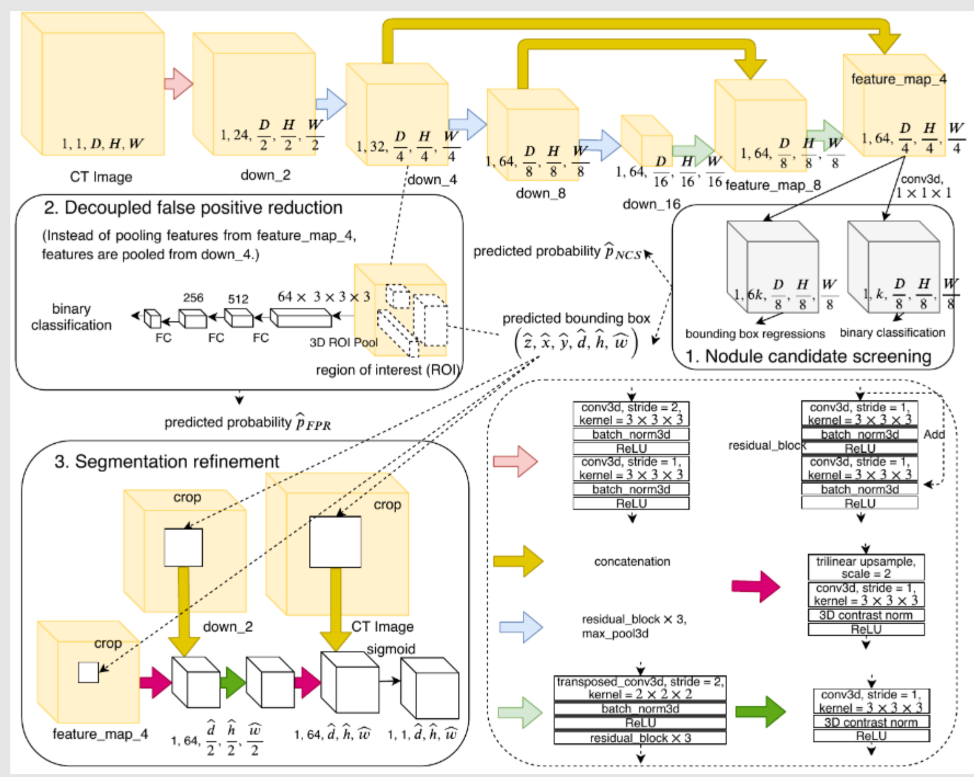

Figure 5. The flow chart of NoduleNet [Tang et al. 2019]

Although the idea for the 3D nodule detection is good and intuitive, there is a big issue to implement the network. The 3D-CNN is highly costly in memory space. Training with GTX3090 (VRAM: 24GB), the program ran out of memory after training 5 images.

References

Stevens, E., L. Antiga, and T. Viehmann (2020). *Deep Learning with PyTorch*. URL: https://pytorch.org/assets/deep-learning/Deep-Learning-with-PyTorch.pdf.

Tang, H., C. Zhang, and X. Xie (2019). "NoduleNet: Decoupled False Positive Reductionfor Pulmonary Nodule Detection and Segmentation". In: *CoRR* abs/1907.11320. arXiv: 1907.11320. URL: http://arxiv.org/abs/1907.11320.

Ronneberger, O., P. Fischer, and T. Brox (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv: 1505.04597 [cs.CV].