Volume R-CNN: Unified Framework for Object Detection and Instance Segmentation in Volumetric Data

Yun Chen[1,2], Junxuan Chen[1] , Bo Xiao[2], Zhengfang Wu[2], Ying Chi[2], Xuansong Xie[2], Xiansheng Hua[2]
[1] Alibaba Group, Hangzhou, China
[2] Beijing University of Posts and Telecommunications, Beijing, China
{chy, xiaobo}@bupt.edu.cn
{junxuan.chen, zhengfang.wz, xinyi.cy, xiansheng.hxs}@alibaba-inc.com
xingtong.xxs@taobao.com

Abstract

As a fundamental task in computer vision, object detection methods for the 2D image such as Faster R-CNN and SSD can be efficiently trained end-to-end. However, current methods for volumetric data like computed tomography (CT) usually contain two steps to do region proposal and classification separately. In this work, we present a unified framework called Volume R-CNN for object detection in volumetric data. Volume R-CNN is an end-to-end method that could perform region proposal, classification and instance segmentation all in one model, which dramatically reduces computational overhead and parameter numbers. These tasks are joined using a key component named RoIAlign3D that extracts features of RoIs smoothly and works superiorly well for small objects in the 3D image. To the best of our knowledge, Volume R-CNN is the first common end-to-end framework for both object detection and instance segmentation in CT. Without bells and whistles, our single model achieves remarkable results in LUNA16. Ablation experiments are conducted to analyze the effectiveness of our method.

1 Introduction

A typical volumetric data is a group of 2D slice images acquired by a CT, MRI, or MicroCT scanner. Usually, these are acquired in a regular pattern (e.g., one slice every millimeter) and have a regular number of image pixels in a regular pattern. For a regular volumetric grid, each volume element (or voxel) is represented by a single value that is obtained by sampling the immediate area surrounding the voxel. The importance of volumetric data multiplies due to the development of the 3D data acquisition field. As a typical volumetric data, CT has proven to be an effective way for early diagnosis. The growth of CT/MRI scanning is around 10–12% per year, but the radiologist workforce grows only 3% per year for the last ten years. This lead to an increase in interpretation error rate by 16.6% because interpretation time is halved (Berlin 2015).

Different from the 2D image field, it is very challenging to fulfill the task of detection on CT due to its characteristic. The target in CT is much tinier than normal objects and it needs several experienced radiologists each spending tens of minutes to draw a convincing conclusion, which makes the CT annotation precious and rare. With tiny target, lack of data and high data dimension, the research on CT is easy to fail due to overfitting, especially when no pretrained models are available because of either commercial confidentiality or diverse data distribution.

Traditional CT diagnosis usually involves hand-designed features or descriptors requiring domain expertise (El-Baz et al. 2011; Murphy et al. 2009). After the large-scale LIDC-IDRI (Armato III et al. 2011) and LUNA16 (Setio et al. 2017) dataset became publicly available, deep learning-based methods have become the dominant framework for nodule research. Current leading methods for CT detection mainly contain two separate steps: propose candidates first and then perform false positive reduction on these candidates with a 3D convolutional neural network (CNN). Dou et al. first established a 3D fully convolutional network (FCN) to screen the candidates from volumetric CT scans, and then a 3D ConvNet classification network is designed to move the false positive candidates (Dou et al. 2017). Ding et al. improved the first stage by introducing 2D RPN to extract proposals in individual 2D images then combine them to generate 3D proposals (Ding et al. 2017). However, these methods are inefficient for both training and inference because candidate proposal and false positive reduction are performed in two separate steps. Worse still, they require sophisticated processing pipeline within the two steps, leading to low efficiency.

We address that candidate proposal and false positive reduction could be joined using RoI Pool methods like RoIMaxPool and RoIAlign, which reduces the number of parameters and computational overhead by sharing convolutional feature maps. This unified system looks like Faster R-CNN (Ren et al. 2015). We further add mask prediction support by introducing a light mask head. The whole system is named Volume R-CNN (see Figure 1), which is a universal detection framework for volumetric data more than CT. In contrast to previous works that rely heavily on handcraft features, specialized knowledge or require complex multi-stage processing, Volume R-CNN is an end-to-end framework could perform object detection and instance segmentation simultaneously and efficiently. Our contributions can be summarized as follows:

1. A novel end-to-end framework (Volume R-CNN) for volume object detection is proposed. It takes 3D volume as input and directly predicts positions, categories and instance mask in 3D space. To the best of our knowledge, it is the first unified and common framework for object

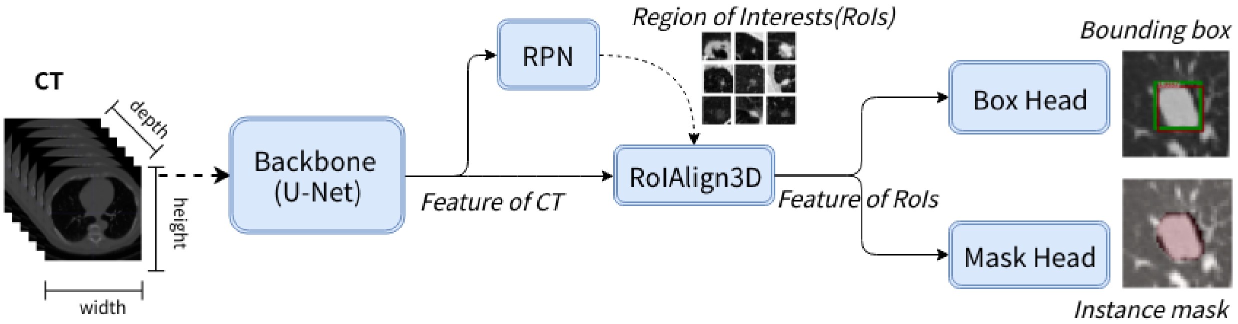


Figure 1: The Volume R-CNN framework for object detection and instance segmentation. RoIs, bounding box and mask are all in 3D space, simplified for visualization herein. Loss from RPN, box head and mask head sum as final train objective. RoIs are seen as input data and the dotted line means no gradient during backward. RoIAlign are they key operation that joins other 4 modules and accelerates the whole process by directly extracting feature of RoIs on the feature map of CT.

detection and instance segmentation in volume. We expected the proposed method could be applied to a wide range of volumetric data and serve as a meta-algorithm for further research in volumetric data.

2. Experimental results have confirmed the effectiveness of our methods. Without bells and whistles, our method could gain competitive results in LUNA16 directly with one single model. Ablation experiments are conducted to investigate the behavior of Volume R-CNN, especially the key component RoIAlign3D. A simplified version of our method has been serving online to process tens of thousands CTs every day.

2 Related Works

Volume R-CNN is a unified R-CNN for volumetric data. In this section, we mainly review the R-CNN family and CT diagnosis systems on behalf of volumetric data. In the 2D computer vision field, R-CNN family work as a masterwork and meta-algorithm for object detection. R-CNN (Girshick et al. 2014) firstly introduced convolutional networks to extract features independently on each RoI and attend to classify a manageable number of candidate object regions. R-CNN was extended in fast R-CNN (Girshick 2015) to allow attending to RoIs on feature maps using RoIPool, leading to fast speed and better accuracy. Faster R-CNN (Ren et al. 2015) advanced this stream by learning the attention mechanism with a Region Proposal Network (RPN) and became the leading framework for object detection. Mask R-CNN (He et al. 2017) further extended Faster R-CNN by adding a branch for mask prediction, as well as replacing RoIPool with RoIAlign which gained a remarkable improvement in mask prediction and became the current meta-algorithm for instance segmentation. Apart from R-CNNs, single stage methods such as SSD (Liu et al. 2016), YOLO(Redmon et al. 2016) are also widely used. Generally speaking, these methods are fast but do not work as accurately as R-CNNs for small objects.

Traditional CT diagnosis involves hand-designed features or descriptors such as morphological features, voxel clustering, and pixel thresholding, requiring domain expertise (Murphy et al. 2009; El-Baz et al. 2011; Aerts et al. 2014; Jacobs et al. 2014; Lopez Torres et al. 2015). Recently, deep ConvNets are employed to generate candidate bounding boxes. Because of the 3D nature of CT data, 3D strategies has shown remarkable advantage (Yan et al. 2016; Shen et al. 2015). Current methods for CT detection are divided into two independent steps by proposing candidates first and then performing false positive reduction (Dou et al. 2017; Ding et al. 2017; Zhu et al. 2018). These methods all consist of two separate steps, which could actually be joined for efficiency. Liao et al. proposed RPN 3D in a lung cancer diagnosis system at (Liao et al. 2017), it is the first known work to direct introduce 3D volume boxes into volumetric data but it is not a generalized implementation and only contains RPN. Apart from CT, there exists some detection methods for point cloud and RGB-D images in self-driving (Song and Xiao 2016; Zhou and Tuzel 2017), which also perform detection in 3D space, but do not work as universal methods for dense volumetric data.

3 The Proposed Method

The proposed method consists of five components, as illustrated in Figure 1. The input is cuboid of size $D \times H \times W$, depth, height, width along the Z, Y, X axes respectively. The backbone is a 3D U-Net extracting features of CT, from which Region Proposal Network (RPN) proposes candidate bounding boxes called region of interests (RoI) — cuboid boxes of different shapes on different locations. The feature of RoIs is extracted using RoIAlign 3D — an efficient module that converts the features inside any valid RoIs with different size into a small feature map with a fixed spatial size. The feature of RoIs is further sent to two relatively independent head to parallelly predict bounding box (*Box Head*) and instance segmentation mask (*Mask Head*) for the target. These components are described detailedly in this section.

3.1 Region Proposal Network (RPN)

A Region Proposal Network (RPN) outputs a set of cuboid object proposals, each with a confidence score. This process is modeled with a fully convolutional network. To generate region proposals, we slide a small network over the convolutional feature map (i.e. $D/4 \times H/4 \times W/4$) output by backbone. The output of RPN is prediction for anchors: a tensor of shape $n \cdot 7 \times D/4 \times H/4 \times W/4$. Here n stands for n anchors on every feature map spatial location, and each anchor has 7 target parameters (6 for location and 1 for confidence score).

Bounding box, Anchor, and RoI Bounding box, anchor and RoI are all cuboid boxes that could be represented by (z_c, y_c, x_c, d, h, w) where z_c, y_c, x_c mean the coordinates of box center and d, h, w refer to the shape. Bounding box usually refers to the ground-truth annotation box or the prediction result that has a valid location and shape. Anchors are predefined boxes of fixed shape (represented by (d, h, w)). Each anchor is shifted in the feature map to generate tens of thousands boxes (z_c, y_c, x_c, d, h, w) — still called as anchor. The output of RPN is the prediction for each shifted anchor. This can be thought of as bounding-box regression from an anchor to a nearby ground-truth bounding box. RoIs refer to region proposals from RPN, which are refined results of anchors with the confidence score.

Batch Sampling For training RPNs, a binary class label (of being an object or not) is assigned to each anchor. We assign a positive label to two kinds of anchors: (i) anchors with the highest Intersection-over-Union (IoU) overlap with a ground-truth box, or (ii) anchors with an IoU overlap higher than 0.5 with any ground-truth box. We assign a negative label to a non-positive anchor if its IoU ratio is lower than 0.02 for all ground-truth boxes. Anchors that are neither positive nor negative do not contribute to the training objective. Only one positive anchor is randomly chosen as the target, and the others do not contribute to the training objective. There are much more negative anchors than positive ones. Hard negative mining (Shrivastava, Gupta, and Girshick 2016) is used to deal with this problem. The N negative samples with highest classification confidence scores are selected as the hard negatives. The others are discarded and not included in the computation of loss. We adopt $N = 2$ in our experiments.

Loss Function RPN give prediction for each anchor $\mathbf{t}_i = (t_z, t_y, t_x, t_d, t_h, t_w, p)$. The target is $\mathbf{t}_i^* = (t_z^*, t_y^*, t_x^*, t_d^*, t_h^*, t_w^*, p^*)$. p is the predicted probability of anchor being an object. The ground-truth label p^* is 1 if the anchor is positive, and is 0 if it is negative. The location parameter is generalized from (Girshick et al. 2014):

$$\begin{aligned} t_{\mathrm{z}} &= (z - z_{\mathrm{a}})/d_{\mathrm{a}}, & t_{\mathrm{d}} &= \log(d/d_{\mathrm{a}}), \\ t_{\mathrm{y}} &= (y - y_{\mathrm{a}})/h_{\mathrm{a}}, & t_{\mathrm{h}} &= \log(h/h_{\mathrm{a}}), \\ t_{\mathrm{x}} &= (x - x_{\mathrm{a}})/w_{\mathrm{a}}, & t_{\mathrm{w}} &= \log(w/w_{\mathrm{a}}), \\ t_{\mathrm{z}}^* &= (z^* - z_{\mathrm{a}})/d_{\mathrm{a}}, & t_{\mathrm{d}}^* &= \log(d^*/d_{\mathrm{a}}), \\ t_{\mathrm{y}}^* &= (y^* - y_{\mathrm{a}})/h_{\mathrm{a}}, & t_{\mathrm{h}}^* &= \log(h^*/h_{\mathrm{a}}), \\ t_{\mathrm{x}}^* &= (x^* - x_{\mathrm{a}})/w_{\mathrm{a}}, & t_{\mathrm{w}}^* &= \log(w^*/w_{\mathrm{a}}). \end{aligned} \tag{1}$$

Here z, y, x, d, h and w denote the box's center coordinates and its shape. Variables z, z_{a}, and z^* are for the predicted box, anchor box, and ground-truth box respectively (likewise for y, x, d, h, w). This can be thought of as bounding-box regression from an anchor box to a nearby ground-truth box. The regression results are predicted boxes.

The loss for RPN is defined by the sum of L_{cls} and L_{reg}:

$$\begin{aligned} L_{rpn}(\mathbf{t}, \mathbf{t}^*) = &\frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \\ &\frac{1}{N_{reg}} \sum_i p_i^* \sum_{k \in \{z,y,x,d,h,w\}} S(t_k^i, t_k^{*i}). \end{aligned} \tag{2}$$

Here, i is the index of an anchor in a mini-batch. t_k^i represents the parameterized coordinate of the predicted bounding box, and t_k^{*i} is that of the ground-truth box associated with a positive anchor. The classification loss L_{cls} is log loss over two classes (object *vs* not object). The regression loss is activated only for positive anchors ($p_i^* = 1$) and is disabled otherwise ($p_i^* = 0$). $S(t, t^*)$ represents the robust loss function smooth L_1 in (Girshick 2015), and a modified version of smooth loss is adopted:

$$S(t, t^*) = \begin{cases} |t - t^*|, & \text{if } |t - t^*| > 1, \\ (t - t^*)^2, & \text{else.} \end{cases} \tag{3}$$

Note that only those sampled anchors contribute to the training objective, and others are discarded.

Proposal Creator RPN produces a prediction for each anchor. Then a non-maximum suppression (NMS) operation with an IoU thresh of 0.2 is performed to rule out the overlapping proposals. The selected location-refined anchors are called Region of Interests (RoI). RoIs are seen as input data to be sent to RoIAlign, and gradient does not backward through them.

3.2 RoIAlign 3D

The RoIAlign 3D operation uses trilinear interpolation (Bourke 1999) to convert the features inside any valid RoIs into a small feature map with a fixed spatial extent of (oD, oH, oW) (e.g., $4 \times 4 \times 4$), where oD, oH and oW are layer hyper-parameters that are independent of any particular RoI. Each RoI is defined by a six-tuple (z, y, x, d, h, w) that specifies its center coordinates and shape.

Previous to RoIAlign, RoIPool serves as a standard operation to extract feature map for RoIs. RoIPool works by dividing the $d \times h \times w$ RoI window into a $oD \times oH \times oW$ grid of subwindows of approximate size $d/oD \times h/oH \times w/oW$ and then usually max-pooling the values in each subwindow into the corresponding output grid cell. The critical problem in RoIPool is quantization problem. RoIPool first quantizes a floating-number RoI to the discrete granularity of the feature map, this quantized RoI is then subdivided into spatial bins which are themselves quantized, and finally, feature values covered by each bin are aggregated by max pooling. Quantization is performed, e.g., on a continuous coordinate x by computing $[x/4]$, where 4 is a feature map stride and $[\cdot]$ is rounding; likewise, quantization is performed when dividing into bins (e.g., $4 \times 4 \times 4$). These quantizations introduce misalignments between the RoI and the extracted features. While this may not impact classifying large objects, which is robust to small translations, it has a significant adverse effect on classifying small objects and predicting voxel-accurate masks, especially for the target in CT which may occupy <10 voxels in feature map after downsampling. To address this, we implemented RoIAlign 3D, 2D version of which was first introduced in Mask R-CNN (He et al. 2017). RoIAlign properly aligns the extracted features with the input by avoiding any quantization of the RoI boundaries or bins (i.e., use $x/4$ instead of $[x/4]$). RoIAlign 3D use trilinear interpolation to compute the exact values of the input features in each RoI bin, which is a straightforward generalization of linear interpolation in 2D.

RoIAlign also brings better forward output and backward gradient, because of the way to computing the feature map in roi bins. For each target voxel in the bin, 8 nearest voxels of the feature map are used to calculate the interpolated value, while for RoIPool, only one voxel is selected after comparison, which results in the bottleneck of gradient backward. For an intuitive understanding of the strength of RoIAlign, we conduct simple experiments and show the results in Figure 2. The output of RoIAlign is much clearer than RoIPool under the same resolution. Also, the gradient of RoIPool tends to be noisy and fuzzy, while the gradient of RoIAlign is much more smooth, balanced and well-proportioned, which indicates that the RoIAlign has superior performance in both forward and backward period. RoIAlign leads to considerable improvements for both box and mask prediction which will be elucidated in the experiments.

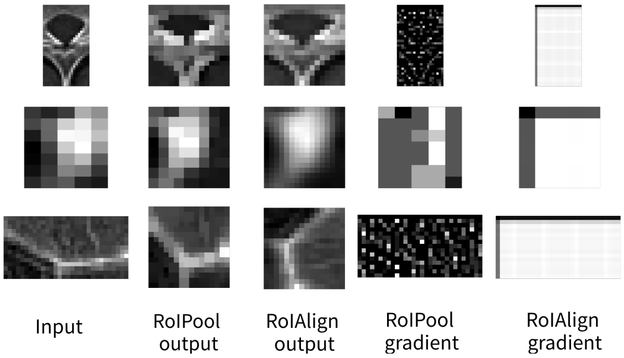


Figure 2: RoIAlign *vs* RoIPool. RoIAlign give better forward output and the gradient is more balanced and well-propotioned in backward. Origin results are 3D cube, center slice is adopted for easy visualization and better understanding.

3.3 Box Head

The RPN emits region proposals without category and the main purpose of the box head is to predict the categories of given RoIs and refine the RoIs to give more accurate location and shape prediction.

Batch sampling We take 8 RoIs from region proposals that have IoU with a ground-truth bounding box of at least 0.3. These RoIs comprise the examples labeled with a foreground object class. 24 RoIs are sampled that have a maximum IoU with ground-truth in the interval $[0.0, 0.001)$, following (Girshick et al. 2014). These are the background examples and are labeled with 0. The sampled RoIs are also used as training target in mask head.

Loss Function The same loss function as Equation 2 is used, except that binary classification is replaced by multi-class classification and location loss is only calculated in the corresponding foreground class. For the one-class detection experiments in this work, the loss function is exactly the same as RPN, because multi-class classification falls back to binary classification. But still, we consider, analyze and implement it as a muti-classification task so that it is applicative to general tasks. Furthermore, it can be seen in the experiments that even for one-class detection task with the same objective, box head still improves results from RPN.

3.4 Mask Head

Mask head gives mask prediction for every RoI. A similar strategy is used for mask representation and training objective as mask R-CNN (He et al. 2017) except that Volume R-CNN works in the 3D space.

Mask Representation For every RoI, mask head gives a mask of $m \times m \times m$, double size of the RoI feature. The ground-truth mask within the bounding box is resampled to the same size. The prediction procedure is addressed naturally by the pixel-to-pixel correspondence provided by convolutions. Specifically, the mask from each RoI is predicted using an FCN. This allows each layer in the mask branch to maintain the explicit object spatial layout without collapsing it into a vector representation that lacks spatial dimensions. This pixel-to-pixel behavior requires RoI features, which themselves are small feature maps, to be well aligned to preserve the explicit per-pixel spatial correspondence faithfully. RoIAlign exactly matches the requirements.

Loss Function The RoIs sampled from box head are also used as the target in mask head. The mask branch has a $K \times m \times m \times m$ dimensional output for each sampled RoI, which encodes K binary masks of resolution $m \times m \times m$, one for each of the K classes. A per-pixel sigmoid is applied and defining L_{mask} as the average binary cross-entropy loss. For a RoI associated with ground-truth class k, L_{mask} is only defined on the k-th mask (other mask outputs do not contribute to the loss). The definition of L_{mask} allows the network to generate masks for every class without competition among classes which decouples mask and class prediction.

3.5 Implementation Details

Network Architecture The detector network consists of a heavy U-Net (Ronneberger, Fischer, and Brox 2015) backbone and 3 relatively light module: RPN head, box head and mask head as is shown in Figure 3.The input of the network is a volume of $128 \times 128 \times 128$. The network backbone is the same as (Liao et al. 2017). U-Net output a feature map of shape $128 \times 32 \times 32 \times 32$. It is followed by two $1 \times 1 \times 1$ convolutions layers with channels 64 and $n \times 7$ respectively, which results in the output of size $7 \cdot n \times 32 \times 32 \times 32$. The 4D output tensor is resized to 5D tensor $7 \times n \times 32 \times 32 \times 32$. The first two dimensions correspond to regressors and anchors

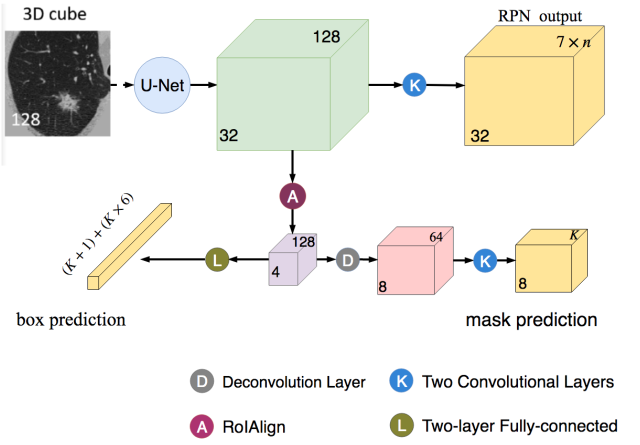


Figure 3: Net architecture of our method. It includes a heavy U-Net backbone the same as (Liao et al. 2017) and 3 light heads (RPN, box head, and mask head). Input is a 3D cube and feature maps are 4D tensors. For convenience, only width and height are presented. The number in the top right of feature map stands for channel. n is number of anchors and K is the number of class

respectively. There are $n \times 32 \times 32 \times 32$ anchor boxes in total, and each has a target of $(t_z, t_y, t_x, t_d, t_h, t_w, p)$. RoIs are proposed by RPN and then RoIAlign extract feature of these RoIs, each producing a feature map of $128 \times 4 \times 4 \times 4$. These feature maps are flattened and sent to a two-layer fully connected layer with 512 hidden units and produce a vector of size $(K + 1) + (K \times 6)$, where K stands for the number of classes and extra 1 for the background. $(K \times 6)$ are the location target for each class. In mask head, the feature map is upsampled using deconvolution followed by normal convolution and produce mask prediction of size $8 \times 8 \times 8$.

Training We used patch-based method, each CT cropped into size of $128 \times 128 \times 128$ cube. Positive and negative patches are randomly selected according to (Liao et al. 2017). The loss from RPN, box head and mask head sum as final target: $L = L_{rpn} + L_{box} + L_{mask}$. As the mask head and the box head are independent of each other, so they can be removed in ablation experiments. We train the model in 4 GPUs with 5 cropped CTs per GPU (so effective mini-batch size is $5 \times 4 = 20$) for 60 epochs with a learning rate of 0.01 which is decayed by 0.1 at the 40 epoch. For the first 10 epoch, we use learning rate *warmup* (Goyal et al. 2017) $lr = 0.01 * epoch^2/100(epoch \leq 10)$. We use stochastic gradient descent (SGD) with a momentum of 0.9 and do not use weight decay.

Inference During inference, the CT is cropped with an overlap to processed separately. At test time, the RPN would generate around 200 region proposals for each cropped CT. We first run the box prediction on these proposals, followed by non-maximum suppression. Instead of predicting masks of RoIs directly, the mask head predicts the mask for refined boxes from box head which is more accurate. The mask branch can predict K masks per RoI, but only the k-th mask is used, where k is the class prediction by the box head. Note that masks are computed only on the top detection boxes that has probability larger than 0.01. Finally, the results are combined and produce results and non-maximum suppression is performed again on the whole CT.

4 Experiments

We perform a thorough comparison of Volume R-CNN to other methods along with detail ablation experiments. LUNA16 (Setio et al. 2017) is adopted in the experiments for comparison and analysis.

4.1 Main experiments on LUNA16

LUNA16 contains 888 chest CT scans and 1186 pulmonary nodules. Each scan, with a slice size of 512×512 voxels, around 0.6 $mm/voxel$, and was annotated during a two-phase procedure by four experienced radiologists. Participants are required to perform 10-fold cross-validation when they use the provided data both as training and as test data. Results are evaluated using the Free-Response Receiver Operating Characteristic (FROC) analysis (on Radiation Units and Measurements 2008) which is defined as the average of the sensitivity at seven predefined false positive rates: $1/8, 1/4, 1/2, 1, 2, 4$, and 8 FPs per scan.

Data Preprocess We adopt similar data preprocess as RPN 3D (Liao et al. 2017). First, the mask of the lung is extracted, and only the lung area is kept, other areas are set to fixes number, followed by intensity normalization. Besides, the location information is also introduced to the network. For each image patch, its corresponding location crop is calculated, which is as big as the output feature map $(32 \times 32 \times 32 \times 3)$. The location crop has 3 channels, which correspond to the normalized coordinates in Z, Y, X axis. Data augmentation is used to alleviate the overfitting problem. The patches are randomly left-right flipped and resized with a ratio between 0.8 $mm/voxel$ and 1.15 $mm/voxel$ during training and 1 $mm/voxel$ during inference. Three anchors of size $(5 \times 5 \times 5)$, $(10 \times 10 \times 10)$ and $(20 \times 20 \times 20)$ are used for this experiment.

Results The result on box head is used as the final prediction but mask head is also included in the training objective. In Table 1, we compare Volume R-CNN to other methods reported in the official conclusion (Setio et al. 2017) of LUNA16 and some claimed leading results in the website (https://luna16.grand-challenge.org/results/). For those publicly available methods, our method gets very competitive results with one single model without bells and whistles. It is notable that other leading methods on the website do not offer the detailed description due to commercial confidentiality and intellectual property, and it may not be a fair comparison.

In Figure 4, we compare our FROC curve with the leading methods reviewed in official report. Our single model gains remarkable superior results than all models in the official report, both in sensitivity and accuracy. It can also be inferred from Figure 4(b) that the train set shows slightly overfitting compared to the result in 10-fold validation but within an acceptable margin.

Table 1: Results in LUNA16

method	FROC
Our Single model (Volume R-CNN)	0.884
DeepLung (Zhu et al. 2018)	0.842
3D FCN+CNN (Dou et al. 2017)	0.839
2D R-CNN+3D CNN (Ding et al. 2017)	0.891
2D SSD (Liu et al. 2016)	0.649
PAtech	0.951
JianpeiCAD	0.950
iFLYTEK-MIG	0.941
iDST-VC	0.897
AIDENce	0.807

The performance of Volume R-CNN could be further promoted by: (i) higher data resolution, as what will be shown in the ablation experiments, remarkable promotion could be achieved when using a higher resolution, but we do not conduct 10-fold validation to report its result for the efficiency. (ii) model ensemble, by combining the results of models, significantly better results can be achieved. These time-consuming tricks do not show much significance and are not applicable to production. More effort is spent on the ablation analysis of Volume R-CNN in this work.

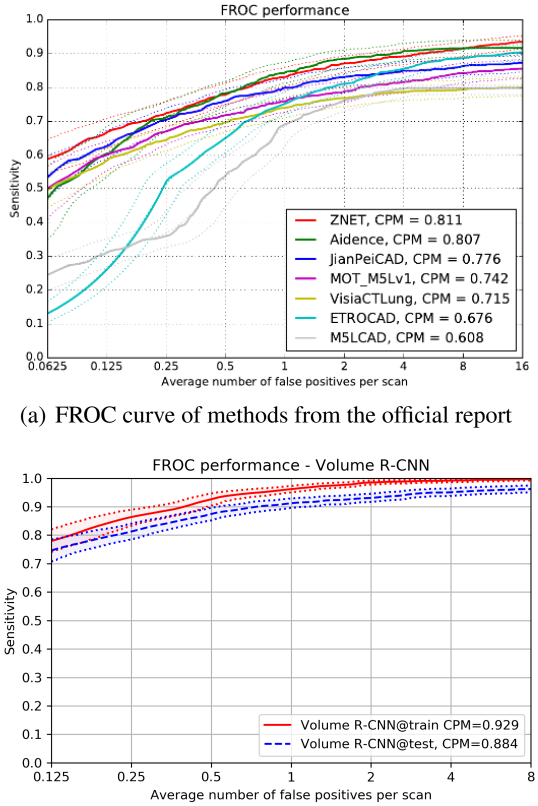
Results visualization Volume R-CNN outputs are visualized in Figure 5. The left is a nodule in the CT visualized in the 3D view. Detecting such a target is as hard as looking for a needle in a sea. We crop the CT at a side length of 36 with the nodule in the center for further visualization. It is notable that each mask is annotated by 4 radiologists and they are averaged as the ground-truth mask. The last row shows some unsatisfying results. One is missed while the other is the false positive. But it is also found that for the false positive, the mask prediction could work as a false positive reduction by stay inactivated on the false positive bounding boxes. But we do not use mask prediction to refine the box results in this work.

Speed and Memory Consumption We trained our model on Nvidia Tesla M40; it occupies around 3GB GPU memory when batch size is 1, 10 GB when batch size is 8. Neither box head nor mask head adds much memory ($\leq$ 200MB for each cropped CT). During inference, it takes around 10 seconds to test the whole CT depending on the size. Compared with RPN, mask head and box head only slightly slow down the speed (less than 10%). The speed and memory consumption could be further optimized by exploration of network architecture, which is left for future work.

4.2 Ablation Experiments

We run several ablations experiments to analyze Volume R-CNN. Results are shown in Table 2 and discussed in detail next. We use 10-fold validation for the fair comparison with other methods in the previous subsection, however, in the ablation experiments, the results are compared within our method, so we used subset 0 as the test set and train on subset 1–9 to accelerate experiments.

Box Head and Mask Head As can be inferred from Table 2, box head (*Box* vs. *RPN*) and mask head (*Mask* vs. *Box*) can both give a promotion to the performance, compared to RPN. This can be interpreted as that box head adds another procedure of classifying to give a more accurate prediction (the same as false positive reduction). While the mask head mainly benefits from more information (mask data) added to guide the training procedure.



(a) FROC curve of methods from the official report

(b) FROC curve of our model

Figure 4: Comparison of FROC. (a) results of previous methods, borrowed from official report (Setio et al. 2017). (b) result of our single model on the train set and validation set.

RoIPool *vs.* RoIAlign RoIAlign gives significant improvement to the model, which can be seen obviously from the comparison of (*Box with Pool* vs. *Box with Align* and *Mask with Pool* vs. *Mask with Align*). Another proof is that *Box With RoIAlign* outperform *Mask with RoIPool*, which demonstrate that even with more data (mask data), RoIPool could not fully utilize them as RoIAlign.

Table 2: Ablation Comparison

method	resolution	Box	Mask	RoI Layer	FROC
RPN 3D	1 mm	-	-	-	0.870
Box with Pool	1 mm	✓	-	RoIPool	0.875
Box with Align	1 mm	✓	-	RoIAlign	0.891
Mask with Pool	1 mm	✓	✓	RoIPool	0.880
Mask With Align	1 mm	✓	✓	RoIAlign	0.905
Box in 0.5 mm	0.5 mm	✓	-	RoIAlign	**0.915**

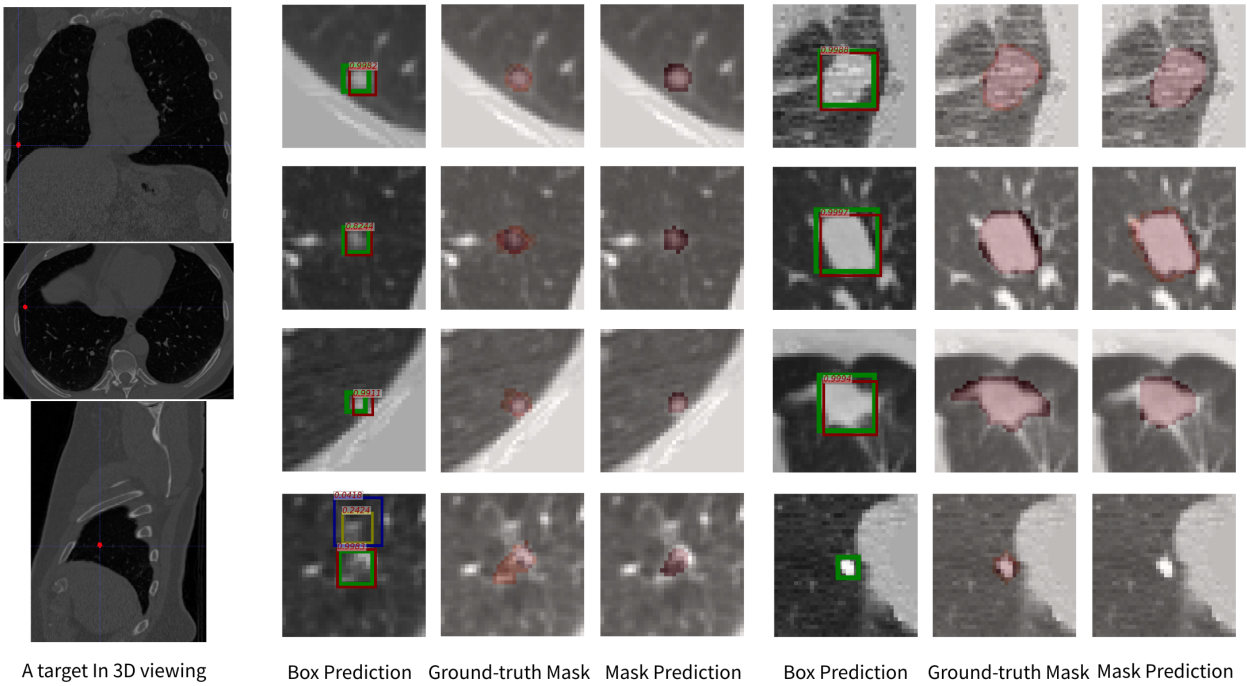


Figure 5: Selected results of Volume R-CNN on LUNA16. Left shows a nodule mask in 3D view (better viewed in color) and others are results of detection. Origin results are the 3D cube. For easy visualization and better understanding, the target is cropped in the center with a side length of 36 voxels and the center slice is visualized. The ground-truth bounding box is drawn with a green box with a thicker edge without probability. The last row shows some unsatisfying results.

Data Resolution Data resolution has a great impact on the performance (0.891 *vs.* 0.915), which seems straightforward. The original data resolution of CT is around $0.6\ mm/voxel$ and after resampled to $1\ mm$, some information is inevitably to lose, which has a crucial impact on the small targets. For example, a nodule with a diameter of $5\ mm$ would only occupy less than 125 voxels (cube with side length of 5 pixels) with resolution $1\ mm/voxels$. If it is resampled to $0.5\ mm/voxel$, it would be a cube with side length 10, occupying around 1000 voxels. Even though higher data resolution gives a great promotion, it is not used online since it greatly slows down the processing (around 8x slower).

5 Conclusion and Future Work

Most of the existing methods in volumetric CT detection require hand-crafted feature, multi-step processing or are confined to specific data. We novelly propose a unified detection framework named Volume R-CNN that joins region proposal, classification and instance segmentation using RoIAlign. It dramatically reduces computational overhead and number of parameter and could be trained end-to-end. Without bells and whistles, our single model gains competitive results on LUNA16. Ablation experiments have been conducted to detailedly analyze the effectiveness of our method. A simplified version of our method has been serving online to process tens of thousands CTs every day.

It is expected that Volume R-CNN to be further applied to more tasks and serve as a meta-algorithm for tasks like lesions registration and tracking. We are also looking forward to those techniques from 2D detection to be migrated to further improve the performance of Volume R-CNN, such as focal loss (Lin et al. 2017b) and FPN (Lin et al. 2017a) .

References

Aerts, H. J.; Velazquez, E. R.; Leijenaar, R. T.; Parmar, C.; Grossmann, P.; Carvalho, S.; Bussink, J.; Monshouwer, R.; Haibe-Kains, B.; Rietveld, D.; et al. 2014. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. *Nature communications* 5:4006.

Armato III, S. G.; McLennan, G.; Bidaut, L.; McNitt-Gray, M. F.; Meyer, C. R.; Reeves, A. P.; Zhao, B.; Aberle, D. R.; Henschke, C. I.; Hoffman, E. A.; et al. 2011. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics* 38(2):915–931.

Berlin, L. 2015. Faster reporting speed and interpretation errors: Conjecture, evidence, and malpractice implications. *Journal of the American College of Radiology* 12(9):894–896.

Bourke, P. 1999. Interpolation methods. *Miscellaneous: projection, modelling, rendering.* (1).

Ding, J.; Li, A.; Hu, Z.; and Wang, L. 2017. Accurate pulmonary nodule detection in computed tomography images using deep convolutional neural networks. In *International*

Conference on Medical Image Computing and Computer-Assisted Intervention, 559–567. Springer.

Dou, Q.; Chen, H.; Jin, Y.; Lin, H.; Qin, J.; and Heng, P.-A. 2017. Automated pulmonary nodule detection via 3d convnets with online sample filtering and hybrid-loss residual learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 630–638. Springer.

El-Baz, A.; Nitzken, M.; Khalifa, F.; Elnakib, A.; Gimel'farb, G.; Falk, R.; and El-Ghar, M. A. 2011. 3d shape analysis for early diagnosis of malignant lung nodules. In *Biennial International Conference on Information Processing in Medical Imaging*, 772–783. Springer.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; and He, K. 2017. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2980–2988. IEEE.

Jacobs, C.; van Rikxoort, E. M.; Twellmann, T.; Scholten, E. T.; de Jong, P. A.; Kuhnigk, J.-M.; Oudkerk, M.; de Koning, H. J.; Prokop, M.; Schaefer-Prokop, C.; et al. 2014. Automatic detection of subsolid pulmonary nodules in thoracic computed tomography images. *Medical image analysis* 18(2):374–384.

Liao, F.; Liang, M.; Li, Z.; Hu, X.; and Song, S. 2017. Evaluate the malignancy of pulmonary nodules using the 3d deep leaky noisy-or network. *arXiv preprint arXiv:1711.08324*.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017a. Feature pyramid networks for object detection. In *CVPR*, volume 1, 4.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017b. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*, 21–37. Springer.

Lopez Torres, E.; Fiorina, E.; Pennazio, F.; Peroni, C.; Saletta, M.; Camarlinghi, N.; Fantacci, M.; and Cerello, P. 2015. Large scale validation of the m5l lung cad on heterogeneous ct datasets. *Medical physics* 42(4):1477–1489.

Murphy, K.; van Ginneken, B.; Schilham, A. M.; De Hoop, B.; Gietema, H.; and Prokop, M. 2009. A large-scale evaluation of automatic pulmonary nodule detection in chest ct using local image features and k-nearest-neighbour classification. *Medical image analysis* 13(5):757–770.

on Radiation Units, I. C., and Measurements. 2008. Receiver operating characteristic analysis in medical imaging. *ICRU Report n 79* 79.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Setio, A. A. A.; Traverso, A.; De Bel, T.; Berens, M. S.; van den Bogaard, C.; Cerello, P.; Chen, H.; Dou, Q.; Fantacci, M. E.; Geurts, B.; et al. 2017. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge. *Medical image analysis* 42:1–13.

Shen, W.; Zhou, M.; Yang, F.; Yang, C.; and Tian, J. 2015. Multi-scale convolutional neural networks for lung nodule classification. In *International Conference on Information Processing in Medical Imaging*, 588–599. Springer.

Shrivastava, A.; Gupta, A.; and Girshick, R. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 761–769.

Song, S., and Xiao, J. 2016. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 808–816.

Yan, X.; Pang, J.; Qi, H.; Zhu, Y.; Bai, C.; Geng, X.; Liu, M.; Terzopoulos, D.; and Ding, X. 2016. Classification of lung nodule malignancy risk on computed tomography images using convolutional neural network: A comparison between 2d and 3d strategies. In *Asian Conference on Computer Vision*, 91–101. Springer.

Zhou, Y., and Tuzel, O. 2017. Voxelnet: End-to-end learning for point cloud based 3d object detection. *arXiv preprint arXiv:1711.06396*.

Zhu, W.; Liu, C.; Fan, W.; and Xie, X. 2018. Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and c lassification. *arXiv preprint arXiv:1801.09555*.