Carnegie Mellon University

Master Thesis

# 3D Object Detection from CT Scans using a Slice-and-fuse Approach

Anqi Yang
Robotics Institute
School of Computer Science

*Thesis Committee*
Aswin Sankaranarayanan, advisor
Srinivasa Narasimhan, advisor
David Held
Jen-Hao Chang

*Technical Report Number: CMU-RI-TR-19-23*

*A thesis submitted in fulfillment of the requirements for the degree of Master of Science in Robotics*

May 2019

# *Abstract*

Automatic object detection in 3D X-ray Computed Tomography imagery has recently gained research attention due to its promising applications in aviation baggage screening. The huge resolution of an individual 3D scan, however, poses formidable computational challenges when coupled with deep 3D convolutional networks for inference. In this thesis, we propose the *slice-and-fuse* strategy — a generic framework to leverage image-based detection and segmentation in high-resolution 3D volumes. We encode the input 3D volumes into multiple slices along XY, YZ, and XZ directions, exploit 2D CNNs to generate 2D predictions, and then fuse 2D predictions to 3D estimation. Using the proposed strategy, we design two 3D object detectors for 3D baggage CT scans. *Retinal-SliceNet* uses a unified, single network to detect target objects from the input 3D CT scans. *U-SliceNet* exploits a two-stage paradigm, first generating proposals using a voxel labeling network and then refining the proposals by a 3D classification network. U-SliceNet generates high-quality segmentation masks along with bounding boxes for target objects. We evaluate the two SliceNets on a large-scale 3D baggage CT dataset for three tasks: baggage classification, 3D object detection, and 3D semantic segmentation.

All of the weapons images are from a DETECT 1000 that is not in a deployed system configuration.

# *Acknowledgements*

First and foremost, I would like to offer my special thanks to my advisor Prof. Aswin Sankaranarayanan. It is his insight, wisdom, and patience that guided me through my master study and research. I could not have imagined having a better advisor for my master's thesis.

I am very grateful to Prof. Kumar Bhagavatula for his insightful suggestions to this work, and Prof. Srinivasa Narasimhan and Prof. David Held for serving on my committee. I really appreciate the valuable suggestions and help from Jen-Hao Chang during the writing of this thesis.

I would like to give my special thanks to our collaborators at IDSS Corp., Mark Caron, Omar AlKofahi, Feng Pan, Duy Dao, and James Connelly. And I would like to thank Hui Zhuo for his help. This work would not be possible without all their support.

Last but far from the least, I would like to thank my parents and grandparents for their deep love and continous support throughout my life.

This work is supported by DHS S&T under Contract Number HSHQDC-17-C-B0020. Any opinions, findings, conclusions or recommendations expressed in this thesis do not reflect the view of DHS S&T or IDSS. All of the weapons images are from a DETECT 1000 that is not in a deployed system configuration.

# Contents

**Abstract** i

**Acknowledgements** ii

**List of Figures** iv

**List of Tables** v

**1 Introduction** 1

**2 Related Work** 5
- 2.1 2D Object Detection . . . 5
- 2.2 3D Object Detection . . . 6
- 2.3 3D Object Detection in Baggage CT . . . 7
- 2.4 3D Baggage CT Dataset . . . 7

**3 Slice-and-fuse Strategy for 3D Object Detection** 9
- 3.1 Slice-and-fuse Strategy . . . 9
- 3.2 Retinal-SliceNet — A One-stage 3D Object Detector . . . 11
- 3.3 U-SliceNet — A Two-stage 3D Object Detector . . . 12

**4 Experiments** 15
- 4.1 SliceNets for Baggage Classification . . . 16
- 4.2 SliceNets for 3D Object Detection . . . 18
- 4.3 U-SliceNet for 3D Segmentation . . . 18

**5 Conclusion** 23

**Bibliography** 24

# List of Figures


1.1 Object detection and segmentation on a 3D baggage CT scan. . . . . 1
1.2 3D baggage CT data. . . . . 2
1.3 Illustration of slice-and-fuse strategy. . . . . 3

3.1 Illustration of slice-operation. . . . . 10
3.2 Retinal-SliceNet architecture. . . . . 11
3.3 U-SliceNet Architecture. . . . . 13

4.1 Results for 3D baggage classification. . . . . 17
4.2 Results for 3D semantic segmentation. . . . . 19
4.3 Qualitative results for object detection and segmentation on Real-scan dataset. . . . . 21
4.4 Qualitative results for object detection and segmentation on Multiple-targets dataset. . . . . 22

# List of Tables

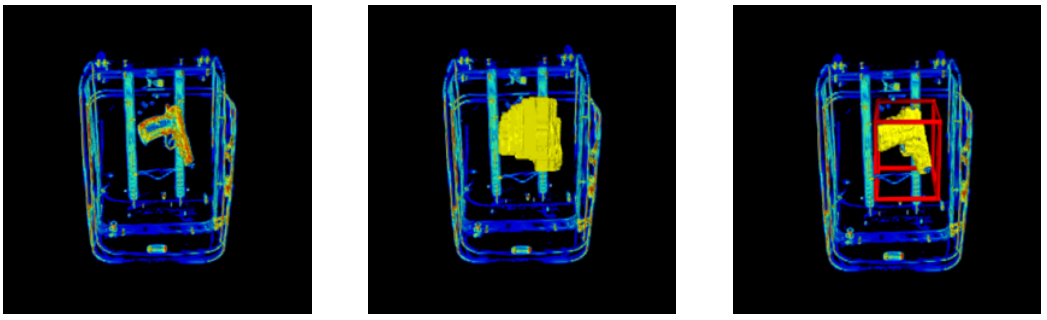
2.1 Subsets description of IDSS 3D baggage CT dataset. . . . . 8

4.1 Results for 3D baggage classification. . . . . 18
4.2 Results for 3D object detection. . . . . 18
4.3 Results for 3D semantic segmentation. . . . . 19

*For my family.*

# Chapter 1

# Introduction

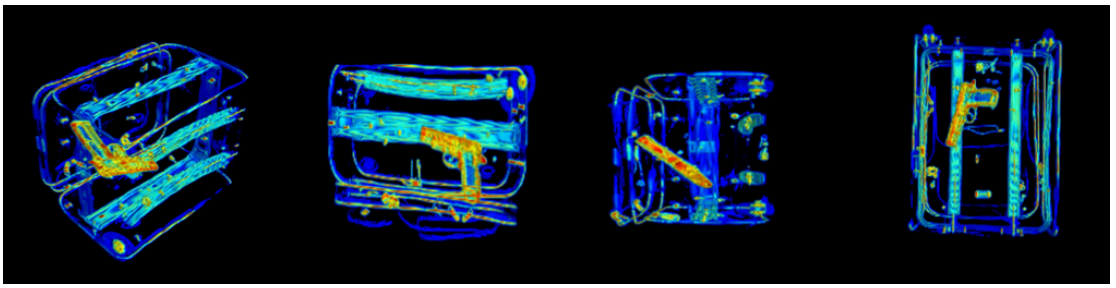


(A) Input baggage CT (B) One-stage SliceNet (C) Two-stage SliceNet

FIGURE 1.1: **Object detection and segmentation on a 3D baggage CT scan.** (a) is the input baggage CT volume of resolution $560 \times 560 \times 560$, (b) shows one-stage SliceNet detection result, (c) shows two-stage SliceNets object result. The yellow masks are predicted target region, and the red box is the predicted target bounding box.

With recent advances in speed and reconstruction accuracy, 3-D X-ray Computed Tomography (3D CT) screening has begun to play a crucial role not only in medical imaging [1, 2], but also in baggage screening for airport security [3–8]. Figure 1.2 shows four viewpoints of a 3D CT scan of a bag. Compared to other 3D scanning techniques, CT scanning has many favorable properties: non-intrusive, capable of high-resolution at sub-millimetre scale, and in full 3D voxel representation that is often occlusion free [9]. This provides numerous unique advantages for highly accurate object detection and segmentation.

Despite the success in understanding medical CT scans, the performance of object detection and segmentation on 3D baggage scans are still far from desirable [6, 7]. This can be attributed to the following three reasons. First, detecting of 3D objects is inherently a hard problem due to the variability of the shapes. This is unlike medical imaging, where intra-class variability is far less and objects are generally of the same shape. Second, baggages are often heavily cluttered with objects such as cell phones, keys, shoes and



(A) Bird-eye-view (B) Side-view (C) Front-view (D) Top-view

FIGURE 1.2: **Different views of a 3D baggage CT scan.** The color represents the density at each voxel. Density increases from blue to red.

etc, many of which have similar shape and density to the target objects. Third, the size of each baggage CT scan is extremely large, often many hundreds of voxels in each dimension. For data of such high resolutions, it is difficult to leverage complex deep neural models while keeping the full 3D resolution due to the limitation of computational resources. For example, implementing a 50-layer ResNet [10] in 3D could need 500GB for $256 \times 256 \times 256$ inputs and a mini-batch of 8. Even if we ignore the storage and memory constraints, the time to train such a deep model would be formidable. If we largely downsample the input volume in spatial resolution at the first few 3D convolutional layers, the small target objects as well as those with thin structures would be eliminated in the subsequent feature extraction and be missed out in the detection results. If we use a 3D fully convolutional network, we need to train it with large 3D reception fields since target objects such as rifles are long in one dimension and placed in the baggage in random pose. This will again lead to insufficient memory problem. How to design an object detection algorithm to achieve high detection accuracy, low training time complexity, and real-time testing speed becomes the key concern.

In this thesis, we propose a novel *slice-and-fuse* strategy that can reduce the computation complexity for object detection and segmentation in high-resolution 3D volumes. Slice-and-fuse works as follows: we first slice the 3D volume into multiple 2D slices; next, we perform detection and segmentation on individual 2D slices and finally pool the 2D predictions in 3D space. Specifically, as is shown in Figure 1.3, in the slicing stage we divide the input volume into multiple 3D slices and project each slice into a 2D image. We repeat this slicing operation along XY, YZ, XZ directions and obtain three sets of 2D images. In the fusion stage, the three sets of 2D predictions are used to reconstruct three 3D volumetric predictions, one for each direction. Among the three volumetric predictions, we select the two most confident predictions for each voxel and fuse them to obtain the final 3D prediction. This 3D prediction seeds subsequent region proposals and classification functions.

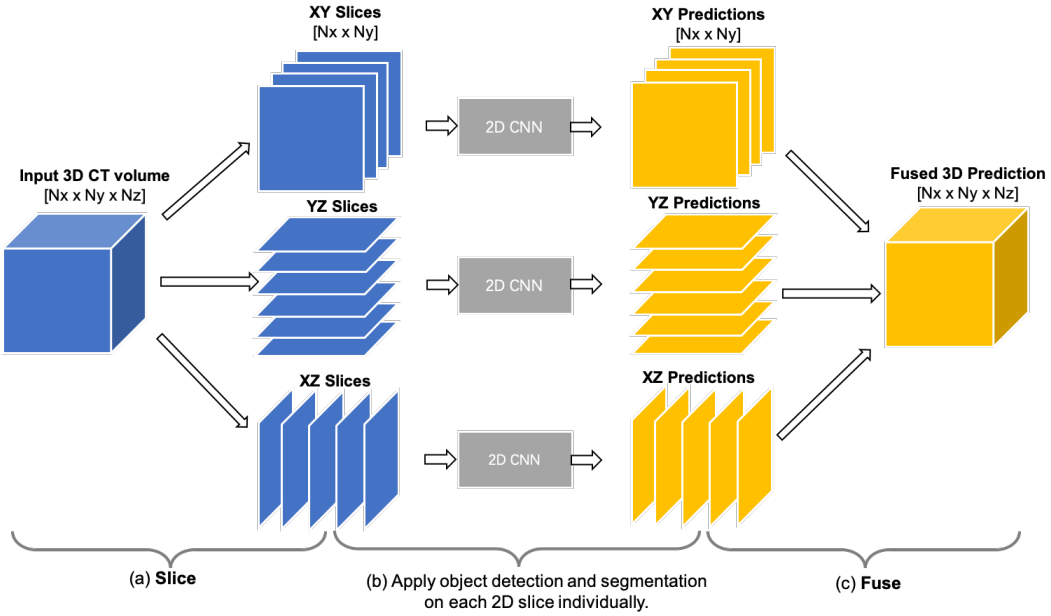



FIGURE 1.3: **Slice-and-fuse strategy.** In order to achieve real-time object detection and segmentation on high-resolution volumes, we first encode the input 3D volume into XY, YZ, XZ slices, apply image-based models on each slice individually, and then pool 2D predictions to 3D space. The proposed strategy can be flexibly incorporated into one-stage and two-stage object detection and segmentation frameworks.

Our strategy is based on two main observations. First, projecting a whole baggage CT scan onto a single 2D plane will cause severe occlusion among target objects and cluttered objects. A simple method to avoid heavy occlusion is to slice a 3D volume into multiple thin 3D slices and then project each of them into a 2D image. The slicing method is especially suitable for object detection task since no matter how large the input volume is, the object detector only focuses on one receptive field at a time. This gives us the flexibility to divide the whole input into multiple slices and perform object detection on each one of them. Second, there exist optimal, sub-optimal as well as non-informative viewpoints when it comes to specific object categories of interest [11]. A pistol, for example, can be easily recognized if both the barrel and grip panel are shown in the projection. By taking advantage of this property, we fuse the two most confident predictions to obtain the final voxel prediction. This not only suppresses the false prediction generated from the confusing viewpoint but also guarantees the prediction consistency among different viewpoints.

To verify the effectiveness of the slice-and-fuse strategy, we propose two 3D object detection networks, which we call *SliceNets*. They incorporate the proposed strategy into two state-of-the-art object detection frameworks - one-stage [12–14] and two-stage object detection [15–19]. Figure 1.1 shows an example result of these two algorithms. Both algorithms start from slicing the input volumes and projecting slices into 2D images. In the one-stage object detector that we call *Retinal-SliceNet*, each slice is used to directly

predict the locations of bounding boxes and corresponding confidence scores. 3D bounding boxes are obtained by the linear fusion operation, and the final predictions are given by the two most confident volumetric predictions. In two-stage object detector that we call *U-SliceNet*, the slice-and-fuse strategy is applied only to the region proposal stage. Specifically, each input slice is given to a 2D-UNet [20] architecture to predict pixel-level labeling, and voxel-level labeling is obtained by the fusion operation. This volumetric labeling is later used for bounding box proposal, followed by a classification network and a location regression network. Note that the two-stage object detection algorithms predict not only the object bounding boxes, but also give accurate voxel-level labeling of the target objects.

**Key contributions.** We conduct pioneering research on object detection using high-resolution 3D baggage CT scans. In particular, this thesis makes the following contributions:

- We propose a slice-and-fuse strategy to boost the speed and accuracy of object detection and segmentation on high-resolution 3D volumes.
- We design Retinal-SliceNet, a one-stage 3D object detector that directly detects target objects for input baggage CT scans.
- We design U-SliceNet, a two-stage algorithm that detects target objects and generates high-quality segmentation masks at the same time for input baggage CT scans;
- We illustrate the performance of the proposed SliceNets in the task of 3D baggage classification, 3D object detection, and 3D segmentation.

**Limitations.** The proposed slice-and-fuse strategy assumes that a 3D object can be confidently classified by an optimal 2D viewpoint. When it comes to objects that lack this property, our strategy would fail to detect them. Triangular pyramid, for example, is hard to be distinguished from square pyramid or triangular prism using a single 2D projection.

**Organization.** This thesis is organized as follows. Chapter 2 discusses some of the key related works in object detection. Chapter 3 introduces the proposed slice-and-fuse strategy, Retinal-SliceNet and U-SliceNet. Chapter 4 demonstrates the performance of SliceNets on baggage classification, 3D detection, and 3D segmentation using a large-scale baggage CT dataset. Finally, we conclude the proposed techniques in Chapter 5.

# Chapter 2

# Related Work

In this chapter, we discuss some of the key related work in 2D and 3D object detection, including those used in the context of CT scans.

## 2.1 2D Object Detection

Current state-of-the-art 2D object detectors can be categorized into one-stage and two-stage detectors.

One-stage object detection algorithms use a single network to regress the bounding box locations and class labels for a fixed set of region proposals on the input image. YOLO [13] divides an input image into $S \times S$ grid cells and uses a unified convolutional network to regress bounding boxes, confidence scores, and class probabilities for each grid cell. The follow-up works YOLO9000 [21] and YOLOv3 [22] further boost the performance. Despite the extremely fast speed, YOLOs miss small objects due to the coarse segmentation to the input images. To compensate objects of different scales, SSD [12] introduces feature pyramids to single-shot object detection, generating anchor boxes of different aspect ratios and scales for each feature map locations. Recently, RetinaNet [14] proposes focal loss to handle the extreme imbalance between the background and target object bounding boxes and achieves the state-of-art detection performances.

Two-stage object detection algorithms first generate a small set of candidate regions and refine the class labels as well as locations of the pre-selected regions. The most representative two-stage object detection algorithm is the R-CNN family [15–17]. Faster R-CNN is the first to introduce Region Proposal Network (RPN) to filter out a large number of background candidates and uses a second network to accurately predict class labels and coordinates for each proposal. Recent two-stage object detection algorithms

follow the same paradigm as Faster-RCNN. To avoid the costly classification applied on each region proposal, R-FCN [23] extracts position-sensitive feature maps and feed them to region proposals to directly compute class scores. FPN [18] improves the detection accuracy by introducing multi-scale feature pyramids into Faster R-CNN. A more recent work Mask R-CNN [19] extends Faster R-CNN framework to instance segmentation and achieves the state-of-art performance. They first detect object bounding boxes, and then crop and segment these regions to get the refined mask.

## 2.2 3D Object Detection

Convolutional Neural Networks (CNNs) have brought significant progress in 3D object detection. Many of the previous works convert point cloud into volumetric representation and generalize CNNs to 3D CNNs for object detection. 3D-FCN [24] exploits a 3D fully convolutional network to predict class labels and bounding boxes locations directly. VoxelNet [25] utilizes 3D convolutional layers to encode the input 3D volumes into multi-channel 2D feature maps and feeds the features to a subsequent detection network. Vote3Deep [26] exploits the sparsity of 3D volume to accelerate the 3D convolution. However, these 3D-based algorithms are extremely expensive when applied to detect objects in high-resolution 3D volumes.

To alleviate the computational complexity of 3D object detection, researchers have made progress in leveraging image-based object detection by encoding 3D data into 2D images. VeloFCN [27] projects the 3D point cloud to front-view to obtain a 2D depth map and then applies a 2D detection network to localize the vehicles. MV3D [28] utilizes the bird-eye-view for region proposals and fuses the features from front-view, bird-eye-view, and RGB images to predict object classes and bounding boxes locations. A more recent work Frustum-PointNet [29] first detects objects on RGB images and extrudes the proposals into 3D frustums, and then applies 3D segmentation and bounding box regression within the 3D frustums.

Despite the high efficiency, these image-based algorithms have certain limitations. Firstly, projecting the whole scene to a certain viewpoint could introduce severe occlusion among target objects and cluttered object if the scene if very crowded. Secondly, all these algorithms all rely on fixed viewpoint for region proposal (bird-eye-view or RGB images), which requires the target objects to remain on the ground. When the target objects pose freely on all three dimensions, this implicit assumption no longer holds.

## 2.3 3D Object Detection in Baggage CT

Existing object detection algorithms [6, 7] in 3D baggage CT scans simplify the object detection problem to a template-based matching problem, where the goal is to determine whether a template object appears in the testing baggage. [6] proposed a 3D SIFT descriptor to describe the keypoints in both template and testing volumes, and then applies a RANSAC driven keypoint match selection to determine whether the template is included in the testing volume. [7] further compares the template matching performance using different keypoint descriptors, including density histogram, density gradient histogram, RIFT [30], and 3D SIFT [6]. However, template-based matching method requires the candidate object in the testing baggage to be exactly the same to the template object, even a small change in shape might cause a mismatch. This could pose a severe threat to aviation security.

One plausible solution to object detection in 3D baggage CT scans is to apply an accurate 3D classifier in a sliding-window approach. Extensive research has been done to improve object classification accuracy on 3D baggage CT volumes [4, 5, 8], however, all existing works exploit hand-crafted features, whose performance is restricted by human priors. [4] extracts density histogram and density gradient histogram for each keypoint and encode them with Bag of visual Words. Support vector machine is used for further classification. The following work [8] improves the classification accuracy by exploiting random forest as feature encoding method. [5] propose a visual cortex model to extract 3D features, which is similar to a deep neural network, however the kernel of each layer is hand-crafted. Moreover, even if these classifiers can be incorporated in a sliding-window approach to detect target objects, the computational complexity could be unaffordable in practice, due to the expensive computation on each 3D window.

## 2.4 3D Baggage CT Dataset

Experimental validation of techniques developed in this thesis is performed on a large-scale 3D baggage CT dataset collected by our collaborators at IDSS corp. The dataset consists of five subsets with 11,375 baggage in total. A detailed description of subsets is shown in Table 2.1. True positive baggage scan contains weapons as well as various cluttered objects, while true negative baggage only contains cluttered stuff. Note that threats in low-cluttered, high-cluttered, and multiple-targets subsets are artificially inserted into each scanned baggage. This results in a very large collection of simulated true positive bags, where each weapon is labeled voxel-wise. Real-scan subset contains

a small collection of true positive baggage, where each threat is physically placed in the bags and scanned together with the cluttered objects.

| | **Features** | **Type** | **#Train** | **#Test** |
|---|---|---|---|---|
| **Low Cluttered** | Each baggage is sparsely filled with cluttered stuff such as shoes, bottles. A weapon is synthetically inserted into each baggage. | Simulated true positive | 2,285 | 287 |
| **High Cluttered** | Each baggage is densely filled with cluttered stuff including electric devices (laptops and cell phones). One weapon is synthetically inserted into the bag. | Simulated true positive | 1,934 | 242 |
| **Multiple Targets** | Multiple weapons are synthetically inserted into each baggage. | Simulated true positive | 548 | 69 |
| **Real Scan** | Weapons are physically placed in each baggage and scanned along with the clutters. | Scanned true positive | 0 | 274 |
| **Clearbag Scan** | Each baggage contains only clutters. | Scanned true negative | 2,847 | 2,889 |
| Total | | | 7,614 | 3,761 |

Table 2.1: Subsets description of IDSS 3D baggage CT dataset.

# Chapter 3

# Slice-and-fuse Strategy for 3D Object Detection

In this chapter, we introduce the proposed *slice-and-fuse* strategy, a general framework for object detection and segmentation in high-resolution 3D volumes, and its applications in objection detection and segmentation.

## 3.1 Slice-and-fuse Strategy

The proposed *Slice-and-fuse* strategy is a generic method for object detection and segmentation in high-resolution dense 3D volumes. We aim to leverage 2D CNNs to accelerate as well as boost the accuracy of 3D object detectors. The key to our strategy are the two operations: the *slice* operation that effectively encodes 3D volumes into 2D images, and the *fuse* operation that decodes 2D predictions to recover volumetric estimation. With these two operations, expensive detection and segmentation computation is only carried out in 2D-space, while only marginal computation is needed to convert between 3D and 2D.

Our method takes in 3D dense volumes of arbitrary size, and output 3D predictions of the same size as the inputs. In the following paragraphs, we will introduce slice operation and fuse operation in detail.

**Slice volumes into multi-view slices.** We generate slices along XY, YZ, and XZ directions. Each slice is generated by first cropping out an $n$-voxel-thick subvolume and then performing max-operator along the $n$ voxels. Specifically, as is shown in Figure 3.1, to generate an XY slice from a dense 3D volume $V \in \mathbb{R}^{N_x \times N_y \times N_z}$, we first crop out a

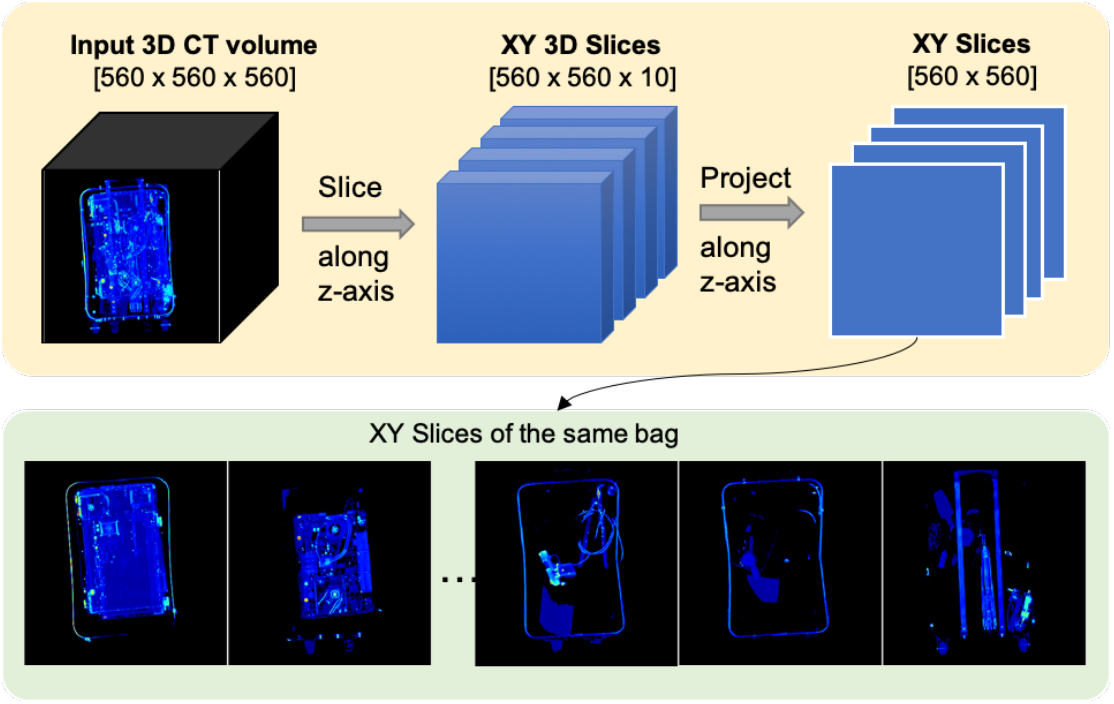



FIGURE 3.1: Illustration of slice-operation along z-axis.

sub-volume of size $N_x \times N_y \times n$ and then apply max-operator along z-axis to get a 2D projection of size $N_x \times N_y$. We apply similar operations to YZ and XZ directions. In all, three sets of slices are generated with respect to the three directions. Each slice is individually fed to 2D image-based CNNs to obtain 2D predictions.

**Fuse multi-view 2D predictions to 3D.** To aggregate XY, YZ, and XZ slices predictions, we first linearly interpolate slices from the same direction to obtain one volumetric prediction for each direction. Specifically, for XY direction, we linearly interpolate $N_z/n$ XY slices along z-axis to recover a 3D prediction $\widehat{V}_{XY} \in \mathbb{R}^{N_X \times N_y \times N_z}$. Similarly we can recover $\widehat{V}_{YZ}$ and $\widehat{V}_{XZ}$ for YZ and XZ directions. To fuse the multi-view 3D predictions, we average the two largest predicted values for each voxel.

Compared to other imaged based 3D object detection algorithms, the slice-and-fuse pipeline offers three advantages for improving detection accuracy. First, cropping out sub-volumes before projection effectively avoids the heavy occlusion caused by projecting the whole volume at one time. Second, since we know the location each slice is extracted from, we retain the depth information. This enables us to accurately estimate full 3D volumes in the fuse operation. Thirdly, averaging the two maximum predictions in the fusion stage explicitly check the spatial consistency across multiple views, making the 3D prediction for each voxel more reliable.

## 3.2 Retinal-SliceNet — A One-stage 3D Object Detector

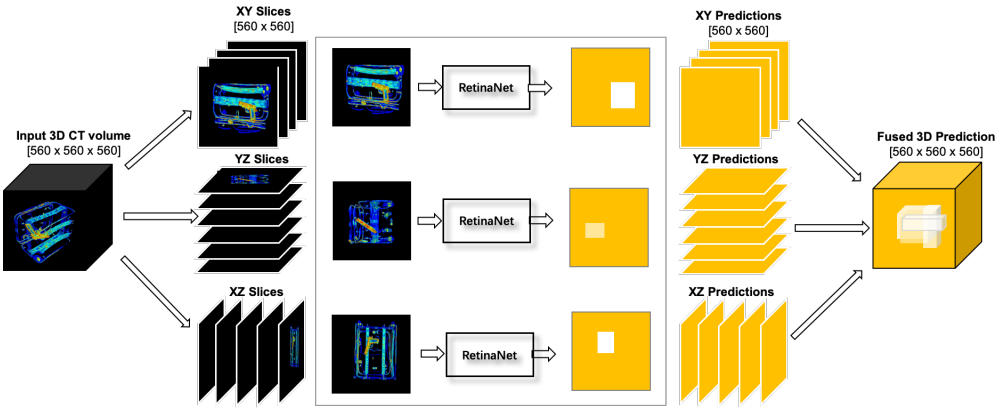



FIGURE 3.2: **Retinal-SliceNet architecture.** Retinal-SliceNet is a one-stage 3D object detector. It incorporates RetinaNet [14] into slice-and-fuse framework to directly predict the location of target objects.

In this section, we present one-stage Retinal-SliceNet, which uses a single network to regress target objects location in 3D baggage CT. The proposed algorithm adopts RetinaNet [14], one of the state-of-the-art one-stage 2D object detection networks, to the proposed slice-and-fuse strategy. Specifically, an input 3D volume is first sliced into XY, YZ, and XZ slices, and each 2D slice is given individually to RetainNet to predict bounding boxes and corresponding confidence scores. Each 2D prediction can be decoded into a continuous score map, where the bounding box regions are assigned with the value of corresponding confidence scores. For example, an input XY image of size $N_x \times N_y$ will produce a predicted continuous score map of size $N_x \times N_y$. In all, three sets of 2D prediction maps can be obtained and fed to the fuse-operation to get a single 3D volumetric estimation of size $N_x \times N_y \times N_z$. The estimated 3D prediction is further thresholded to get the final 3D estimation.

RetinaNet [14] takes feature pyramid network [18] as backbone to extract features from the input image and construct a feature pyramid. For each pyramid level, they attach two small FCNs to it - a classification subnet that predicts the confidence score of each object class, and a box regression subnet that regresses the bounding box locations. The challenge in training RetinaNet [14] for baggage CT slices is the severe imbalance between the number of voxels that are target objects and that are the cluttered background. For baggage that contains one target object, in most cases, only one slice out of more than 15 slices that contains a target. And in a slice, a target object only takes a small portion of the total image resolution.

To address the extreme class imbalance problem, we adopt focal loss [14] during training, which is shown to prevent the vast number of easy negative samples to overwhelm the

detector from training. Focal loss [14] is defined as:

$$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t)$$

where

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases}$$

Focal loss adds two modulating factors $\alpha_t$ and $(1-p_t)^{\gamma}$ to the cross entropy loss. $\gamma \geq 0$ is the focusing parameter. The larger $\gamma$ is, the less loss is affected by the well-classified samples. $\alpha_t$ is the balance parameter. Tuning $\alpha_t$ according to the ratio of positive and negative samples can slightly improved the accuracy.

## 3.3 U-SliceNet — A Two-stage 3D Object Detector

A key component of two-stage object detector is the region proposal network, which coarsely selects regions that contains target objects and proposes a series of anchor boxes. The region proposals are further given to an accurate classification network to refine the class labels.

The proposed U-SliceNet follows the two-stage detector paradigm. In addition to predict bounding boxes, our SliceNet also regresses an accurate segmentation for the target objects. The 3D semantic segmentation comes naturally with our region proposal network, which first predicts accurate voxel-wise labels and then generate anchor boxes. In the following paragraphs, we will describe the region proposal network (voxel-wise labeling followed by region proposal) and the classification network in detail.

**Voxel-wise labeling network.** The voxel-wise labeling network is designed by adapting a simple 2D-UNet [20] to the proposed slice-and-fuse strategy. The labeling network takes in a dense 3D volume of size $N_x \times N_y \times N_z$ and outputs a 3D prediction of the same size, with each voxel represents the probability of belonging to the target objects. Specifically, given a 3D baggage CT volume, we first slice it into XY, YZ, and XZ slices, and feed each slice individually to a 2D-UNet [20]. We use fuse-operation to aggregate all 2D pixel-wise labeling into 3D voxel-wise prediction. The 3D volumetric prediction is further thresholded to keep only the most probable regions. The 2D-UNet has eight fully convolutional layers `conv1-conv8`, followed by eight deconvolution layers `deconv1-deconv8`, with a skip connection connecting each pair of them. In the down-sampling pathway, each `convi` (`i`=1,...,8) layer outputs a feature map with a spatial resolution of $2^i$ lower than the input image and $32 * 2^i$ feature channels. Each conv

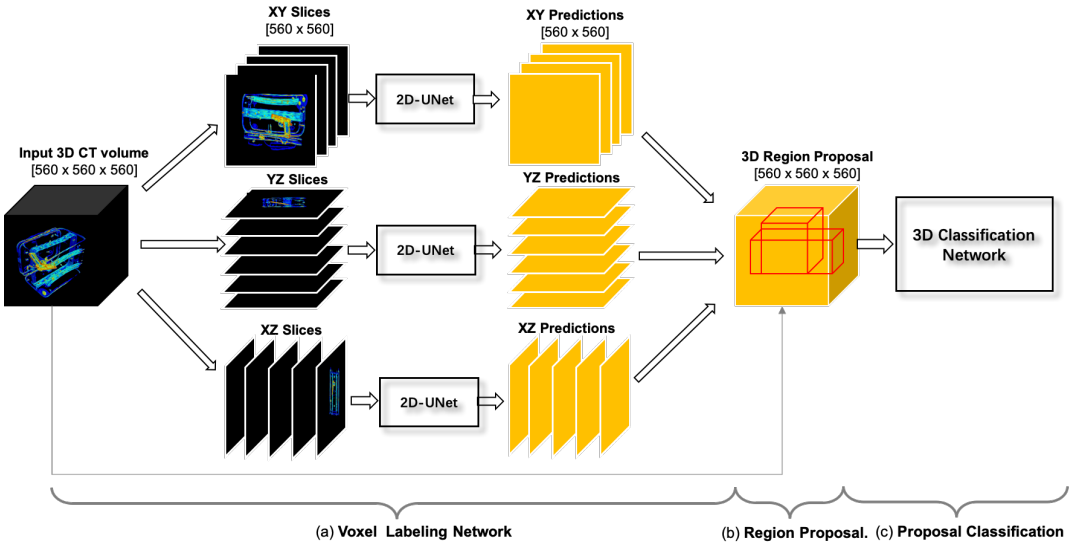



FIGURE 3.3: **U-SliceNet Architecture.** U-SliceNet is a two-stage object detector. In the first stage, voxel-labeling network estimates a 3D volumetric prediction and generates proposals at the valid voxels. In the second stage, a 3D classification network accurately classifies the region proposals.

layer is followed by a LeakyReLu ($\alpha = 0.2$) activation and a batch normalization. The bottleneck feature has a resolution of $1 \times 1$. In the upsampling pathway, each `deconvi` (`i`=8,...,1) layer outputs a feature map with a spatial resolution of $2^{(i-1)}$ lower than the output image, and the final has the same number of channels as the number of object classes. Each `deconvi` (`i`=8,...,2) layer is followed by an ReLU activation and a batch normalization layer.

During the training of 2D-UNet, we use both slices containing target objects and clear slices. Specifically, for the positive baggage, we only select slices containing target objects; for the negative baggage, we randomly select nine slices from each baggage. To handle the class imbalance between target objects and the cluttered background, we use focal loss as training criterion.

**Region proposal.** We select the anchor points with a spatial interval of $m$ voxels among the valid voxels. At each anchor point, we propose a set of anchor boxes that center around the anchor point. To achieve accurate detection, the anchor boxes are designed to have 5 different scales and 8 different aspect ratios, yielding a maximum of 155 anchors at each location. The region proposals are then back-projected to input volumes to crop out corresponding sub-volumes, which are later batch-resized to a unified resolution of $32 \times 32 \times 32$ for further classification.

**Proposal classification.** We train a fully 3D CNN to reject the region proposals that only have cluttered background or small overlapping with ground-truth target objects. The network has three convolutional blocks followed by two fully connected layers. Each convolutional block is composed of four 3D convolution layers, each followed by an ReLU activation. conv1 of each block outputs a feature map with spatial resolution $2^i$ lower than the input 3D feature map and $64 \times 2^i$ number of feature channels. And `conv2-4` of each block keep the sptial resolution and number of channels the same.

# Chapter 4

# Experiments

We experiment the proposed SliceNets on IDSS 3D Baggage-CT dataset. We train our models on the Low-clutter, High-clutter, Multiple-targets and Clearbag datasets, and test on all five datasets including Real-scan dataset. The detailed training and testing splits are summarized in Table 2.1. In all, 7,614 bags are used in training.

**Training set generation.** In order to train image-based component of our SliceNets, we prepare the 2D training samples in advance. Specifically, for target baggage, we select at most nine slices from each baggage that contains the target objects. The slices are selected by first finding the centroid of target objects and generating 3 XY slices, 3 YZ slices, and 3 XZ slices around it. For the clear baggage, we randomly pick nine slices from each baggage. This results in a training set of 47,835 target slices and 25,624 clear slices.

**Retinal-SliceNet implementation details.** The training of Retinal-SliceNet involves only the training of [14]. One Nvidia TITAN Xp GPU is used during training. The input images are of size $560 \times 560$. The model is trained with a mini-batch size of 8 images for 200 epochs. We use a pretrained FPN [18] as backbone network. We adopt SGD for optimization with a learning rate of 0.0001, a weight decay of 0.0001 and momentum of 0.9. For the focal loss function, we use $\gamma = 5$ and $\alpha = 0.25$. By applying the trained RetinaNet in slice-and-fuse strategy, we obtain a 3D continuous prediction. We further threshold the 3D prediction to keep only regions with high possibility to be target objects and give a bounding box to each connected regions.

**U-SliceNet implementation details.** During the training of 2D-UNet [20], each input image is resized to $256 \times 256$. We use one GPU with a mini-batch of 16 images.

We adopt Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ and train the network for 200 epochs. The initial learning rate is 0.002 and is updated every 50 epochs according to lambda update rule. We adopt focal loss as training criterion, with $\gamma = 2$ and $\alpha = 0.25$.

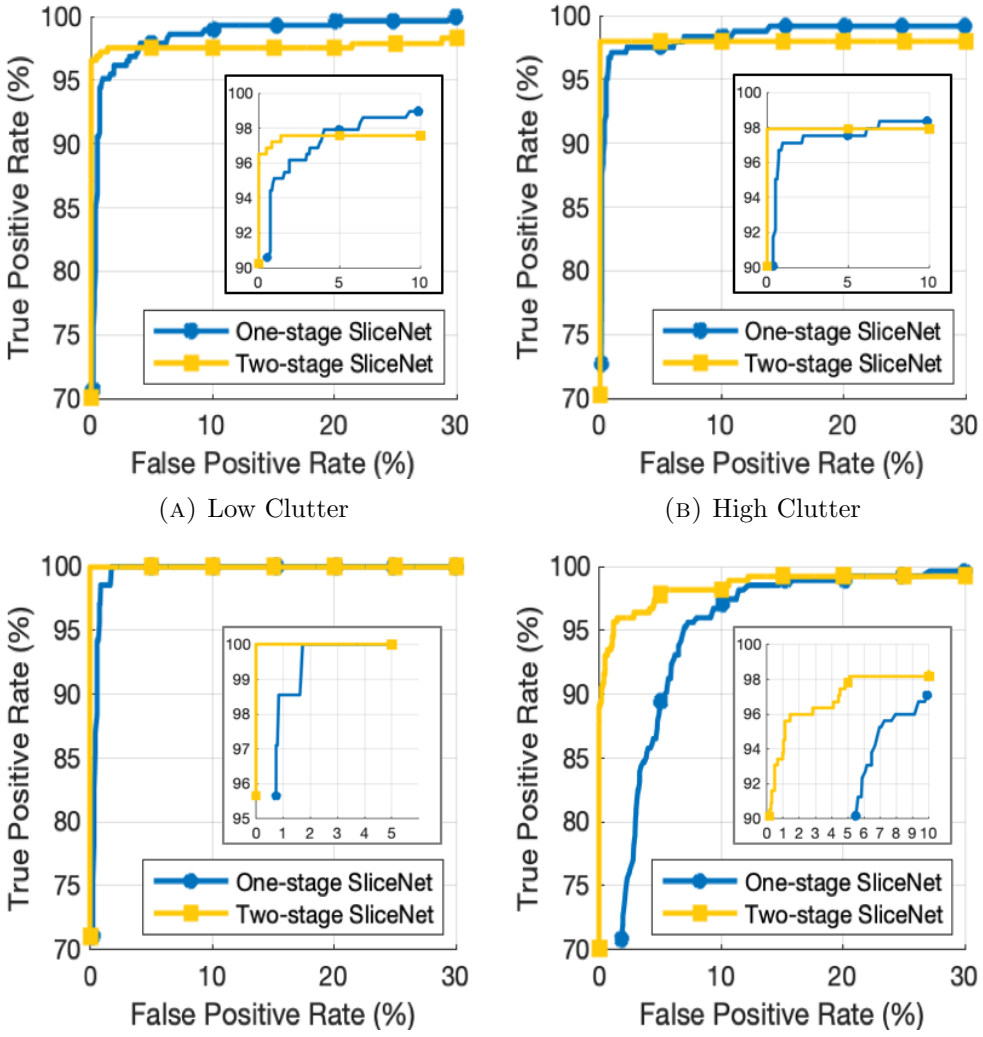
After fusing 2D predictions to 3D estimation, we use a threshold $r_1$ to filter out regions that have small probabilities to be target objects. Among the kept regions, we select anchor points using a spatial interval of 15 voxels, and propose a fixed set of 155 anchors at each location.

We train the 3D classification network on the region proposals generated from training baggage. At each iteration, the network takes in a mini-batch of 4 bags and randomly selects 100 negative proposals and all positive proposals for training. The training labels are determined according to the Intersection-over-Union (IoU) ratios with the ground-truth bounding box as in [17]. The proposals that have IoU greater than 0.4 are assigned as positive samples, and those have IoU less than 0.1 are assigned as negative samples. 4 GPUs are used in training. We use a SGD optimizer with learning rate of 0.0001 and momentum of 0.9 and train for 15 epochs. A focal loss with $\alpha = 0.75$ and $\gamma = 2$ is used as training criterion. We adopt a hierarchical training strategy. We first pretrain the network on 1000 randomly selected training samples for 10 epochs to initialize the large number of weights in the 3D network. Each epoch takes around 20 minutes. And then we train the model on the whole training dataset for five epochs. Each epoch takes around 3 hours.

## 4.1 SliceNets for Baggage Classification

We first show how SliceNets can be used for 3D baggage classification. Baggage that contains weapons are considered a positive sample, and clear baggage is considered a negative sample. We first use SliceNets to predict a score for each bag and then threshold the score to obtain the binary label. For one-stage SliceNet, we consider the largest predicted value of the 3D prediction as the bag-level score. For U-SliceNet, we use the largest bounding boxes score as the bag-level score.

We test the baggage classification accuracy of SliceNets on four target datasets and the clearbag dataset. The ROCs are shown in Figure 4.1. We further summarize the detection rate and false alarm rate in Table 4.1. The recalls and false alarm rates are computed by thresholding the predicted bag-level scores. Two-stage SliceNet is more accurate than one-stage SliceNet in terms of recall and false alarm rate. Note that for Real-scan dataset, which is not used for training, one-stage SliceNet achieves 95.26% detection rate at a false alarm rate of 6.95%, while two-stage SliceNet achieves 98.18%




(A) Low Clutter (B) High Clutter

(C) Multiple Threat (D) Real Scan

FIGURE 4.1: **Results for 3D baggage classification.** For Real-scan dataset, one-stage SliceNet achieves 95.26% detection rate at a false alarm rate of 6.95%, while two-stage SliceNet achieves 98.18% detection rate at a false alarm rate of 5.08%.

detection rate at a false alarm rate of 5.08%. For simulated datasets, two-stage SliceNet achieves a 0% false alarm rate for all three datasets, with a detection rate of 96.52% for the High-clutter dataset, 97.93% for the High-clutter dataset and that of 100% for the Multiple-targets dataset. The reason that Multi-Targets dataset achieves very high detection rate is that in terms of baggage classification task, a sample is determined as positive if at least one weapon is detected. This makes the detection in Multi-Targets baggage easier.

| | | Low Clutter | High Clutter | Multiple Targets | Real Scan |
|---|---|---|---|---|---|
| One-stage SliceNet | Recall (%) | 95.12 | 97.11 | **100.00** | 95.26 |
| Two-stage SliceNet | | **96.52** | **97.93** | **100.00** | **98.18** |
| One-stage SliceNet | False Alarm Rate (%) | 0.98 | 0.98 | 1.69 | 6.95 |
| Two-stage SliceNet | | **0.00** | **0.00** | **0.00** | **5.08** |

TABLE 4.1: **Results for 3D baggage classification.**

## 4.2 SliceNets for 3D Object Detection

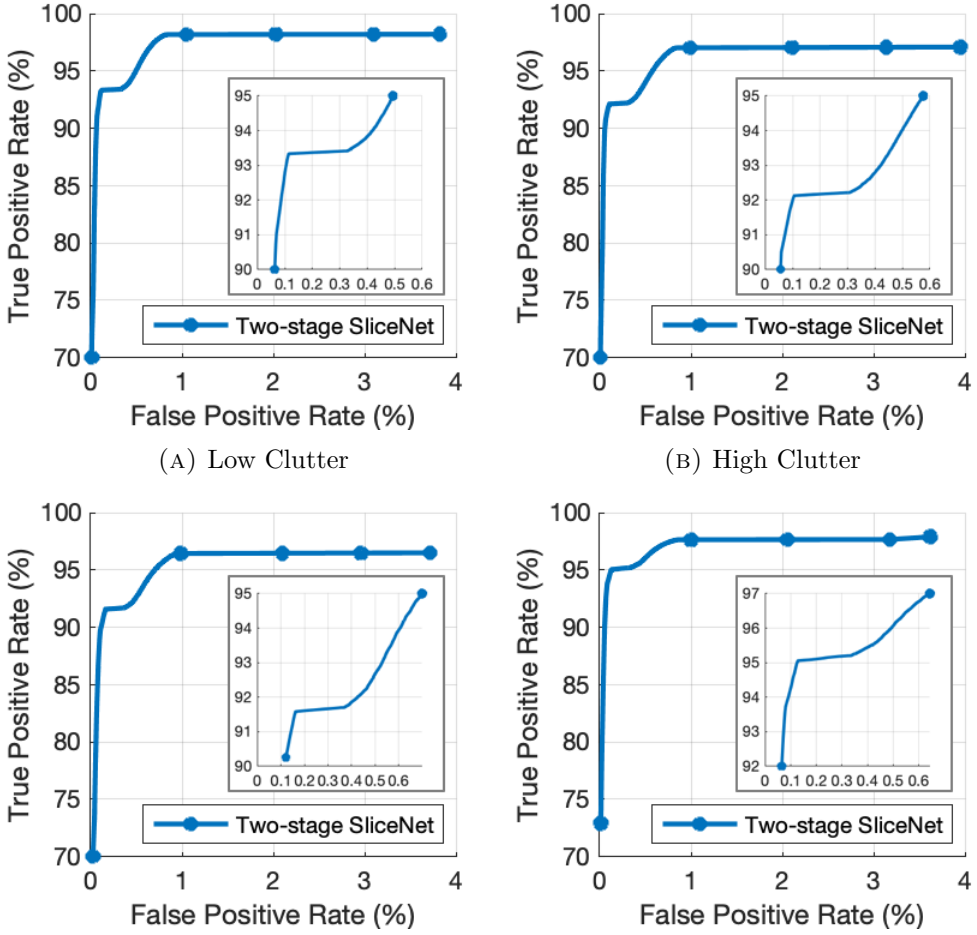
We evaluate two-stage SliceNet on 3D object detection task. We test the average precision of the estimated bounding box under three different thresholds. The results are shown in Table 4.2. We also demonstrate qualitative results in Figure 4.3 and Figure 4.4. We use Average Precision in 3D ($AP_{3D}$) as evaluation metrics, together with an Intersection over Union (IoU) threshold. If the predicted bounding box has an IoU>=threshold with the ground-truth bounding boxes, we consider the bounding box as correct detection. Note that our two-stage SliceNet achieves high accuracy with IoU$\geq$0.3, and the performance drops when IoU threshold increases. This is because we didn't add bounding box regression to the second-stage. Our network can detect the target object with a high accuracy, but are not able to predict the accurate positions. This can be improved by adding bounding box regression in the future work.

| | | Low Clutter | High Clutter | Multiple Targets | Real Scan |
|---|---|---|---|---|---|
| Two-stage-SliceNet | $AP_{3D}$ (IoU=0.3) | 96.11 | 99.16 | 100.00 | 90.51 |
| Two-stage SliceNet | $AP_{3D}$ (IoU=0.4) | 83.04 | 86.97 | 95.65 | 76.28 |
| Two-stage SliceNet | $AP_{3D}$ (IoU=0.5) | 51.59 | 55.04 | 71.01 | 42.70 |

TABLE 4.2: **Results for 3D object detection.**

## 4.3 U-SliceNet for 3D Segmentation

Lastly, we show that U-SliceNet can be used in 3D semantic segmentation task as well. We test it on each of the four target dataset. We predict a score between 0 and 1 for each voxel to indicate the probability of the voxel belongs to a target object. The ground




(A) Low Clutter (B) High Clutter

(C) Multiple Threat (D) Real Scan

FIGURE 4.2: **Results for 3D semantic segmentation.**

truth for each voxel is a binary label for whether being background or targets. Figure 4.2 shows the ROC of evaluating 3D voxel-labeling on each dataset.

By applying a threshold to each voxel prediction, we are able to compute voxel-wise recall and false alarm rate, as is shown in Table 4.3.

| | Low Clutter | High Clutter | Multiple Targets | Real Scan |
|---|---|---|---|---|
| Recall (%) | 93.30 | 92.12 | 91.52 | 95.05 |
| False Alarm Rate (%) | 0.11 | 0.10 | 0.16 | 0.13 |

TABLE 4.3: **Results for 3D semantic segmentation.**

Figure 4.3 and Figure 4.4 showcase three qualitative detection and segmentation results for the Real-scan dataset and Multiple-targets dataset respectively. Each baggage is projected to top view, side view, and front view. Images in the same row are the

projections belong to the same baggage. It can be seen that the two-stage SliceNet is able to generate high quality segmentation masks for objects of different scales and for multiple instances.

Given a $560 \times 560 \times 560$ input volume, two-stage SliceNet takes an average of 5.09s to generate a highly accurate segmentation mask using one Nvidia TITAN Xp together with Intel Xeon CPU E5-2640. We feed each slice to 2D-UNet one at a time to obtain all 2D predictions for a baggage. We implement the linear interpolation between slices in parallel using GPU to reconstruct 3D prediction from 2D predictions.

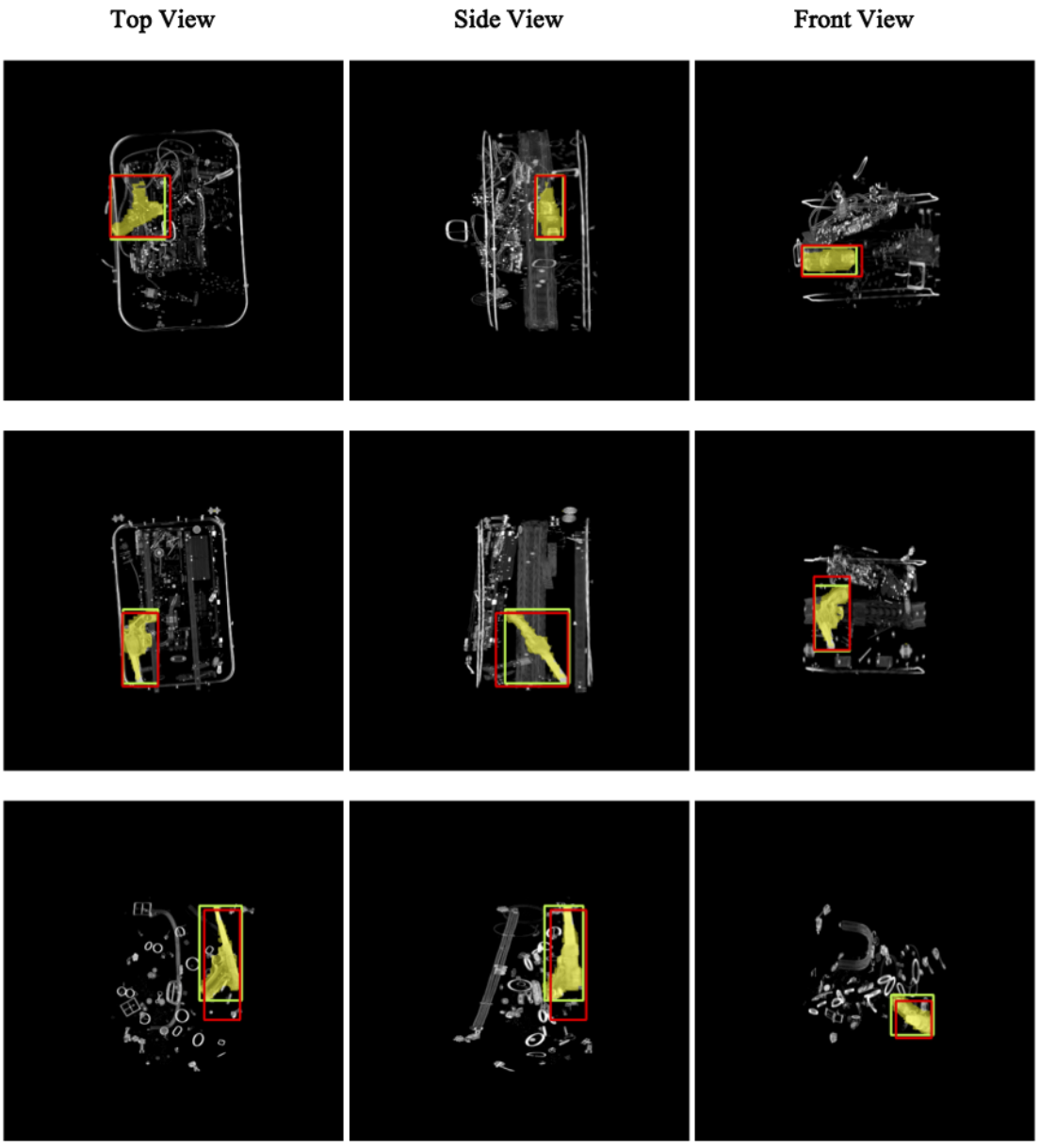



FIGURE 4.3: **Qualitative results for object detection and segmentation on Real-scan dataset.** The three images in each row represents the top-view, side-view, and front-view of a scanned baggage. The original image is shown in gray scale. The red boxes are predicted bounding boxes, and the yellow region are predicted segmentation mask. The green boxes are the ground-truth boxes.

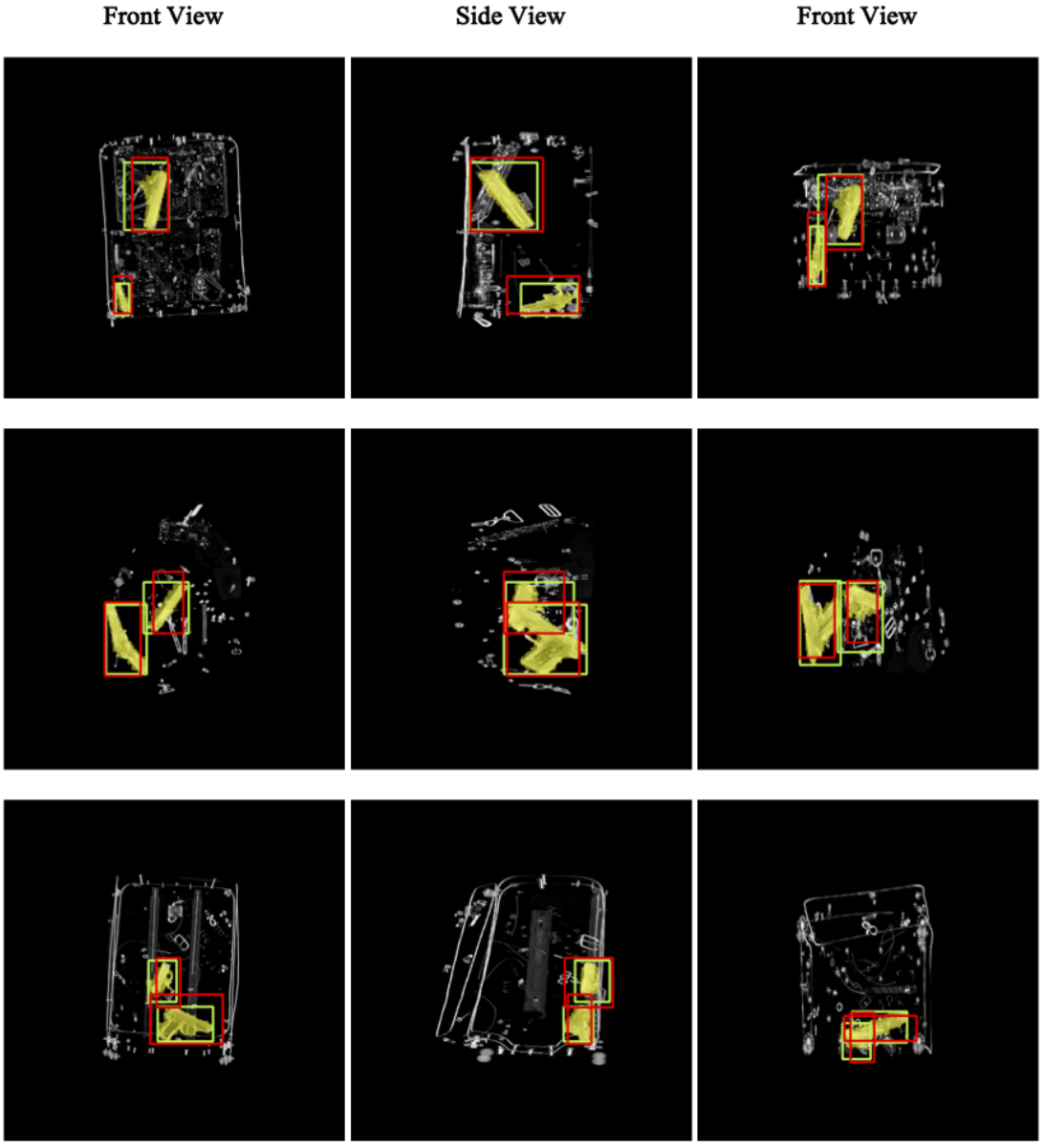



FIGURE 4.4: **Qualitative results for object detection and segmentation on Multiple-targets dataset.** The three images in each row represents the top-view, side-view, and front-view of the same baggage. The original image is shown in gray scale. The red boxes are predicted bounding boxes, and the yellow region are predicted segmentation mask. The green boxes are the ground-truth boxes.

# Chapter 5

# Conclusion

In this thesis, we present *slice-and-fuse* strategy, a generic framework for object detection and segmentation in high-resolution 3D volumes. Our framework encodes input 3D volumes into multiple 2D slices, leverages fast image-based models to produce 2D detection or segmentation results, and then pools 2D predictions to 3D space to obtain volumetric predictions.

We demonstrate slice-and-fuse strategy in object detection for 3D baggage CT scans. We design two algorithms, called *SliceNets*, that incorporate cutting-edge one-stage and two-stage detection algorithms into the proposed strategy. Retinal-SliceNet exploits a single image-based detector [14] to generate 2D predictions and directly fuses the 2D predictions into volumetric predictions. U-SliceNet first leverages 2D-UNet [20] to predict voxel-wise labeling for the input volume, and then assigns anchor boxes to valid voxels. A highly-accurate 3D classifier is trained to refine the classification of region proposals.

We evaluate our SliceNets in baggage classification, object detection on IDSS 3D baggage CT dataset. We also test U-SliceNet in 3D semantic segmentation. For Real-scan dataset, Retinal-SliceNet achieves 95.26% detection rate at a false alarm rate of 6.95%, while U-SliceNet achieves 98.18% detection rate at a false alarm rate of 5.08%. In object detection task, U-SliceNet reaches more than 90% average precision (with IoU$>0.3$) on all four target datasets. However, the performance drops when higher IoU is required. This can be improved by designing bounding box regression in the future work. In semantic segmentation task, the U-SliceNet achieves around $\sim 0.15\%$ false alarm rate and produces visual compelling results.

# Bibliography


[1] Sardar Hamidian, Berkman Sahiner, Nicholas Petrick, and Aria Pezeshk. 3d convolutional neural network for automatic detection of lung nodules in chest ct. In *Medical Imaging 2017: Computer-Aided Diagnosis*, volume 10134, page 1013409. International Society for Optics and Photonics, 2017.

[2] Kamal Jnawali, Mohammad R Arbabshirani, Navalgund Rao, and Alpen A Patel. Deep 3d convolution neural network for ct brain hemorrhage classification. In *Medical Imaging 2018: Computer-Aided Diagnosis*, volume 10575, page 105751C. International Society for Optics and Photonics, 2018.

[3] Rushil Anirudh, Hyojin Kim, Jayaraman J Thiagarajan, K Aditya Mohan, Kyle Champley, and Timo Bremer. Lose the views: Limited angle ct reconstruction via implicit sinogram completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6343–6352, 2018.

[4] Greg Flitton, Andre Mouton, and Toby P Breckon. Object classification in 3d baggage security computed tomography imagery using visual codebooks. *Pattern Recognition*, 48(8):2489–2499, 2015.

[5] Greg Flitton, Toby P Breckon, and Najla Megherbi. A 3d extension to cortex like mechanisms for 3d object class recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3634–3641. IEEE, 2012.

[6] Gregory T Flitton, Toby P Breckon, and Najla Megherbi Bouallagu. Object recognition using 3d sift in complex ct volumes. In *BMVC*, number 1, pages 1–12, 2010.

[7] Greg Flitton, Toby P Breckon, and Najla Megherbi. A comparison of 3d interest point descriptors with application to airport baggage object detection in complex ct imagery. *Pattern Recognition*, 46(9):2420–2436, 2013.

[8] Andre Mouton, Toby P Breckon, Greg T Flitton, and Najla Megherbi. 3d object classification in baggage computed tomography imagery using randomised clustering forests. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5202–5206. IEEE, 2014.

[9] Andre Mouton. On artefact reduction, segmentation and classification of 3d computed tomography imagery in baggage security screening. 2014.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5010–5019, 2018.

[12] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[16] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[21] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[22] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[23] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[24] Bo Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518. IEEE, 2017.

[25] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[26] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361. IEEE, 2017.

[27] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016.

[28] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[29] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[30] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.