SliceNets — A Scalable Approach for Object Detection in 3D CT Scans

Anqi Yang[1], Feng Pan[2], Vishwanath Saragadam[1], Duy Dao[2],
Zhuo Hui[1], Jen-Hao Rick Chang[1], Aswin C. Sankaranarayanan[1]
[1] Carnegie Mellon University, [2] IDSS Corporation

Abstract

One of the most promising approaches for automated detection of guns and other prohibited items in aviation baggage screening is the use of 3D computed tomography (CT) scans. However, automated detection, especially with deep neural networks, faces two key challenges: the high dimensionality of individual 3D scans, and the lack of labelled training data. We address these challenges using a novel image-based detection and segmentation technique that we call the slice-and-fuse framework. Our approach relies on slicing the input 3D volumes, generating 2D predictions on each slice using 2D Convolutional Neural Networks (CNNs), and fusing them to obtain a 3D prediction. We develop two distinct detectors based on this slice-and-fuse strategy: the Retinal-SliceNet that uses a unified, single network with end-to-end training, and the U-SliceNet that uses a two-stage paradigm, first generating proposals using a voxel labeling network and, subsequently, refining the proposals by a 3D classification network. The networks are trained using a data augmentation approach that creates a very large training dataset by inserting weapons into 3D CT scans of threat-free bags. We demonstrate that the two SliceNets outperform state-of-the-art methods on a large-scale 3D baggage CT dataset for baggage classification, 3D object detection, and 3D semantic segmentation.

1. Introduction

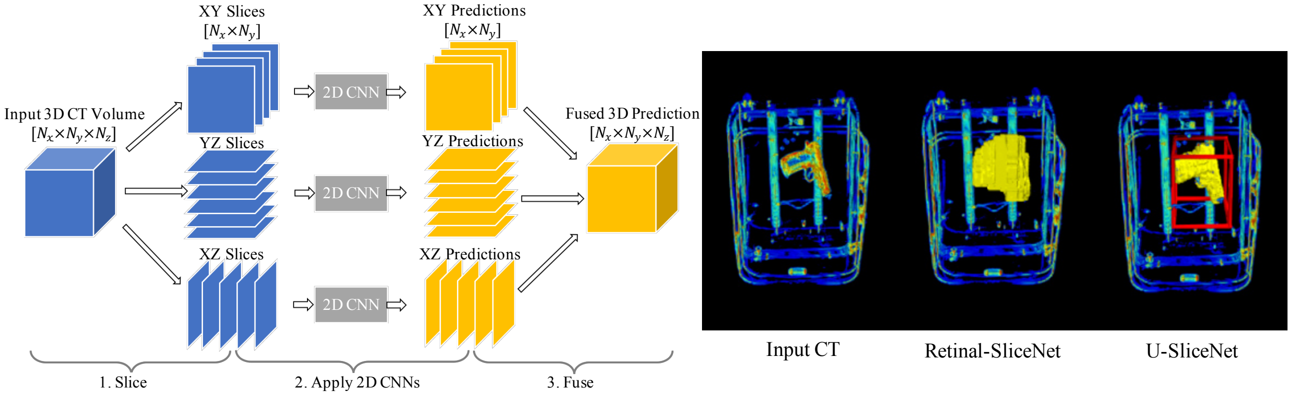
Computed tomography (CT) has many favorable properties over other 3D scanning techniques: non-intrusive, capable of high-resolution at sub-millimeter scale, and largely occlusion free scans [25], which enables highly accurate object detection and segmentation. This has led to successful usage in medical diagnostics [1] as well as detection of threats such as guns and knives [9, 7, 8, 26, 6].

Despite the success of deep learning in many computer vision tasks, there are few results that use deep neural networks to detect objects from CT baggage scans. The primary reason for this can be attributed to the very high-dimensionality of individual 3D scans. For example, a typical scan that we deal with in this paper has a dimensionality of $560 \times 560 \times 560$; for data of such a high resolution, it is difficult to leverage complex deep neural networks, due to computation and memory constraints.

Recent work in 3D object detection has concentrated on point clouds [18, 42, 5, 19, 2, 29, 30, 32, 28, 41], but unfortunately these techniques are not easily applied to dense 3D data. A typical point cloud in KITTI 3D dataset [10] contains $\sim$10K points, while a typical luggage CT scan contains $\sim$175M voxels. Moreover, cluttered background is easier to handle in point clouds as neighboring objects in point clouds are usually separate; in contrast, in baggage scans target objects are often inlaid with others in close proximity. We empirically show that these disadvantages of point cloud detection methods yield unsatisfactory performance on dense 3D data.

There are many strategies [40, 38, 3, 35, 37] that are commonly used for training neural nets on high-dimensional data. The most successful among them applies fully-convolutional networks (FCN) on portions of the training data [3, 35, 37], thereby alleviating the need to process the high-dimensional volume in a single instance. While this strategy works well with compact objects, it is not particularly effective for largely two-dimensional objects like guns, knives and rifles, which have thin elongated structures; the randomness of the pose of such objects naturally requires networks with large 3D reception fields. For the same reasons, strategies that rely on downsampling the volume [40, 38] tend to eliminate thin, but distinctive, structures like the knife blade and the gun barrel.

We adapt a *slice-and-fuse* strategy [36, 16, 14, 27] to weapon detection and segmentation in high-resolution 3D baggage scans that significantly reduces the computation complexity while maintaining a high detection accuracy. As is shown in Figure 1(a), the slice-and-fuse strategy is comprised of three distinct steps. First, a 3D scan is *sliced* into multiple 2D slices along all three cardinal directions. Second, each 2D slice is individually processed using a deep network with the goal of 2D object detection and segmentation. Third, the processed 2D slices are *fused* back into a 3D volume for subsequent post-processing. This strategy is



(a) Slice-and-fuse strategy.

(b) Object detection and segmentation on a 3D baggage CT scan.

Figure 1. **Proposed slice-and-fuse strategy.** (a) Slice-and-fuse works by encoding the input 3D volume into XY, YZ, XZ slices, applying image-based CNN models on each slice individually and extruding 2D predictions to 3D space. (b) Detection and segmentation on a 3D CT scan of a bag. The yellow masks are predicted target region, and the red box is the predicted target bounding box.

based on two main observations. First, guns and knives, the two object categories of interest, are easily recognized from their 2D silhouettes, and hence information in a 2D slice is ample for detection. Second, working with 2D slices allows us to train complex and accurate models, without any of the computational and memory bottlenecks present in 3D processing pipelines.

Contributions. This paper provides a novel approach for detection of guns and knives from 3D CT scans and makes the following contributions.

- *[Retinal-SliceNet]* We design a one-stage object detector based on the RetinaNet architecture [21] for detecting bounding boxes around weapons.
- *[U-SliceNet]* We design a two-stage algorithm based on the UNet architecture [23] that performs semantic segmentation on the 2D slices. The segmentations are used to propose 3D regions for classification.
- *[Data augmentation using threat insertion]* We use a data augmentation strategy wherein 3D scans of isolated weapons are inserted into threat-free bags. This provides a large dataset for training that produces remarkable improvements in detection performances.

The contributions above are tested on a real-scanned dataset containing guns and sharp objects. At a false alarm rate of 5%, the proposed strategy produces detection rates upwards of 98.71% for guns and 61.27% for sharps, taking as little as five to ten seconds per bag.

2. Prior Work

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

2D Object Detection. Existing 2D object detectors can be categorized into one-stage and two-stage detectors. One-stage object detection techniques use a single network to estimate the bounding box locations and class labels for a fixed set of region proposals on the input image. YOLO [33] divided an input image into grid cells and used a unified convolutional network to regress bounding boxes and class probabilities for each grid cell. To handle objects of different scales, SSD [22] introduced feature pyramids as well as anchor boxes of different aspect ratios and scales for each feature map location. More recently, Lin et al. [21] proposed RetinaNet that utilized focal loss to handle the extreme imbalance between the background and target object bounding boxes, which led to the state-of-art detection performance in object detection.

Two-stage techniques first generate a small set of candidate regions and subsequently refine the class labels as well as locations of these regions. The most representative two-stage object detection algorithm is the R-CNN family of techniques [12, 11, 34, 13]. In this context, Faster R-CNN approach proposed by Ren et al. [34] introduced the concept of a region proposal network for filtering out a large number of background candidates; it then used a second network to accurately predict class labels and coordinates for each proposal. The work by Lin et al. [20] improved the detection accuracy further by introducing multi-scale feature pyramids into Faster R-CNN.

3D Object Detection. Convolutional Neural Networks (CNNs) have brought significant progress in 3D object detection. Most of the previous works including 3D-FCN [18], VoxelNet [42] and Vote3Deep [5] converted point cloud into volumetric representation and generalized CNNs to 3D CNNs for object detection. However, these 3D-based algorithms are extremely expensive when applied directly to detect objects in high-resolution 3D volumes.

To alleviate the computational complexity of 3D object detection, a common approach is to encode and process the 3D data as a 2D image. VeloFCN [19] projected the 3D point cloud to obtain a 2D depth map and then applied a 2D detection network to localize vehicles. AniProb [31] used enlongated kernels to encode a 3D volume to a 2D slice and then performed object detection on the 2D slice. Chen et al. [2] proposed utilizing a bird-eye-view for region proposals and fused the features from front-view, bird-eye-view, and RGB images to predict object classes and bounding boxes locations. Frustum-PointNet [29] first detected objects on RGB images and extruded the proposals into 3D frustums for subsequent segmentation. Despite their high efficiency, the use of a fixed viewpoint for region proposal and projection of the entire volume is tuned specifically to sparse depth maps. When applied to dense 3D CT scanes, such approaches face the problem of severe occlusion between target objects and cluttered background.

Object Detection in Point Cloud. More recent work [30, 32, 28] resorts to the sparse and irregular feature of point clouds and directly detects 3D objects in raw point clouds. PointNet [30] took n points as input and extracted global and local features and outputed per point scores. PointNet++ [32] recursively applied PointNet on learned partitions of input point sets. [15, 39] further exploit local dependency by dividing a point cloud into slices or parallel beams and fusing local features to a final segmentation mask. More recently, VoteNet [28] and its successor MLCVNet [41] achieve the state-of-the-art results. They first extracted a subset of "seed" points to generate votes for the center of target object, and then clustered these votes to obtain 3D bounding boxes. These methods implicity assumed that objects are placed seperately. However in highly cluttered baggage scans, target objects are tightly inlaid with or even inside other objects, and thus pose additional chanllenges in accurate object detection.

3D Object Detection in CT Scans. Despite the success of deep neural networks in 3D point clouds, few methods use deep neural networks to detect and segment objects in baggage CT scans. Most existing object detection algorithms [9, 7] simplified the detection problem to a template-based matching problem. For example, a 3D extension to the classical SIFT descriptor [24] and matching was used in [9] to detect objects in the CT scan. This approach was extended in [7] for other keypoint descriptors, including density histogram, density gradient histogram, and RIFT [17]. However, template-based matching can be extremely slow when we need to match candidate objects against multiple target objects and diverse poses. One plausible solution to object detection in 3D baggage CT scans is to apply an accurate 3D classifier in a sliding-window approach. [8, 26, 6] applied classifiers to hand-crafted feature descriptors such as density histogram and density gradient histogram, and led to sub-optimal performance. Moreover, using a sliding window is often computationally expensive, precluding processing of very large volumes.

Closely related to our work, [36, 16, 14, 27] exploited 2D slices and view aggregation to segment medical CT scans. Our work is different from them in two significant ways. First, we focus on the application of weapon detection in baggage CT; Second, while these work tackle semantic segmentation task, we extend their applicability to 3D object detection and classification.

3. SliceNets

The proposed *slice-and-fuse* strategy leverages 2D CNNs to accelerate object detection and segmentation in high-resolution dense 3D volumes. Our strategy relies on two key operations: the *slice* operation that effectively encodes 3D volumes into a collection of 2D images, and the *fuse* operation that decodes 2D predictions to recover volumetric estimation. With these two operations, the most computationally intensive components — namely, the learning-based formulation for detection, and segmentation — are only carried out in the 2D space, while the rest of the processing of the 3D data is computationally lightweight. In the following paragraphs, we will introduce slice operation and fuse operation in detail.

The slicing operation. Slicing generates a set of 2D slices for a single 3D scan. Suppose that the three axes of the 3D scan are denoted using X, Y, and Z. Slicing along the Z direction produces a collection of XY slices. 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 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 Y and X directions to get a collection of XZ and YZ slices. This generates in total $N_x/n + N_y/n + N_z/n$ 2D slices when there is no overlap between two slices in one direction.

Processing each slice. Each slice is then individually processed using a 2D image-based CNN to obtain 2D score maps in terms of object/segmentation labels. Further details will be described in Sections 3.1 and 3.2. Once we obtain predictions at each slice, we apply the fusing operator to construct a 3D volume. Note that the 2D image-based CNN is the only component that needs to be trained in the proposed framework, which significantly reduces computational and memory burden. In the training step, we generate 2D slices and their labels by slicing the density volumes and corresponding ground-truth volumes.

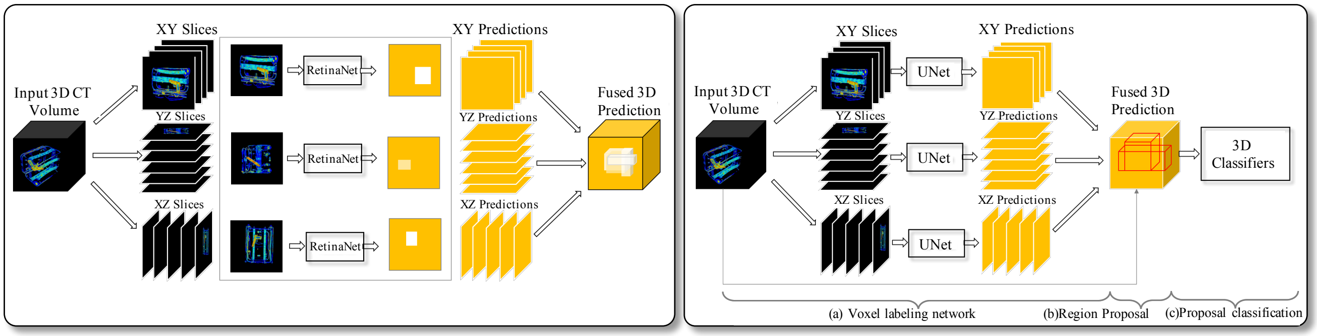


Figure 2. **SliceNet architectures.** (left) Retinal-SliceNet is a one-stage 3D object detector. It incorporates RetinaNet [21] into slice-and-fuse framework to directly predict the location of target objects. (right) U-SliceNet is a two-stage object detector with 3D volumetric segmentation followed by a 3D classification network.

The fusing operation. 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 construct a 3D prediction $\widehat{V}_{XY} \in \mathbb{R}^{N_x \times N_y \times N_z}$. Similarly we construct $\widehat{V}_{YZ}$ and $\widehat{V}_{XZ}$ for YZ and XZ directions. To fuse the multi-view 3D predictions, we average the k largest values from three directions voxel by voxel, where k is a hyper-parameter for different classes.

Advantages of the slice-and-fuse strategy. Compared to other image-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 into a single 2D image. Second, since we know the location each slice is extracted from, we retain the 3D voxel information. This enables us to accurately estimate full 3D volumes in the fuse operation. Third, averaging all three or the two maximum predictions in the fusion stage explicitly ensures spatial consistency across multiple views, making the 3D prediction for each voxel more reliable.

The slice-and-fuse strategy can be easily incorporated into state-of-the-art pipelines for 3D object detection and segmentation. We show two distinct approaches based on one-stage and two-stage detection pipelines.

3.1. Retinal-SliceNet — A One-Stage Detector

Retinal-SliceNet, adopted from the RetinaNet architecture [21], uses a single network to regress target objects locations. As shown in the left part of Figure 2, after producing XY, YZ and XZ slices from an input 3D volume, each slice is processed individually using a RetinaNet that predicts bounding boxes and their corresponding confidence scores. Each 2D prediction can be decoded into a continuous score map, where the bounding box regions are filled with the value of corresponding scores. After obtaining all 2D scores maps from three directions, we feed them to the fuse-operation to get a single 3D volumetric estimation, which is further thresholded to be the final 3D estimation. We defer the details of 2D object detection to [21], and provide implementation details in the supplementary material.

3.2. U-SliceNet — A Two-Stage Detector

As shown in the right part of Figure 2, the proposed U-SliceNet follows the two-stage detector paradigm and consists of three components: a voxel-wise labeling network that operates over 2D slices, a 3D region proposal module on fused 3D labels, and a subsequent classifier of the proposed 3D regions.

Voxel-wise labeling network. The voxel-wise labeling network is designed by adapting the 2D-UNet [23] to the proposed slice-and-fuse strategy. Each 2D slice is then processed using a standard UNet architecture with the objective of producing pixel-level labeling. After the fusing operation, we obtain a coarse 3D voxel-labeling for each volume.

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 detections, 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.

Proposal classification. We train one 3D CNN to select region proposals that have large overlaps with target guns and one for target sharps. The network takes in cropped sub-volumes from region proposal stage and resizes them to a unified resolution of $32 \times 32 \times 32$. The locations of selected proposals are used as predicted bounding boxes; the max score among all region proposals for each class is used as the bag-level score for that class. We further apply the selected proposals to the coarse voxel-labeling and generate a final semantic segmentation mask for target objects.

4. Dataset

While our SliceNet architectures address the formidable computational challenges posed by the high-dimensionality of CT scans, training the underlying model still requires a large training dataset. Collecting such datasets is extremely time consuming as we not only need to assemble a plethora of distinct bags, but also painstakingly label them to get voxel-level segmentation masks.

We address the challenges in assembling a large dataset using a simple and effective data augmentation strategy inspired by [4]. Our strategy relies on having access to a large number of CT scans of weapons-free bags, which can be easily obtained from the stream of commerce, i.e., bags that pass through the checkpoint at an airport. Once we have these bags, we can take CT scans of weapons in isolation, rotate them and insert them into free spaces in the bags. This provides us with a rich dataset where the weapons are inserted into a rich set of backgrounds. Further, we can transfer the segmentation mask of the weapon into the final bag to get a precise voxel-level labeling.

With this data augmentation procedure, we build a mix of real and simulated threat dataset for training and evaluation of our approach. We organize this dataset into multiple subsets, which are described in Table 1. On the whole, we assembled a dataset that has 1001 real scans of bags with weapons in them during scanning, and nearly $15,700$ simulated threat bags where we digitally insert weapons. In addition, we collected a set of $11,400$ threat-free bags from the stream of commerce.

Our data augmentation strategy has some limitations. We can only insert weapons into bags that have ample clear spaces and so the training data has lesser background clutter. To address this, we include a portion of the weapons-free bags as negative examples in the training process. A second limitation is that a single weapon could potentially be inserted in multiple bags (although under different rotations) and hence, the threats themselves are correlated. This is a point of concern for articulated objects like scissors that are likely to occur in many configurations.

5. Experiments

We evaluate the proposed SliceNets on 3D Baggage-CT dataset for three different tasks — baggage classification, 3D object detection, and 3D semantic segmentation. We train our models on simulated datasets as well as 10% bags from the clearbag dataset, and test on all five simulated datasets and two real-scan datasets.

Training set generation. In order to train SliceNets, we prepare the 2D training samples in advance. Specifically, for target baggage, we select nine slices around each target object from each baggage. 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 118,790 target slices and 11,879 clear slices.

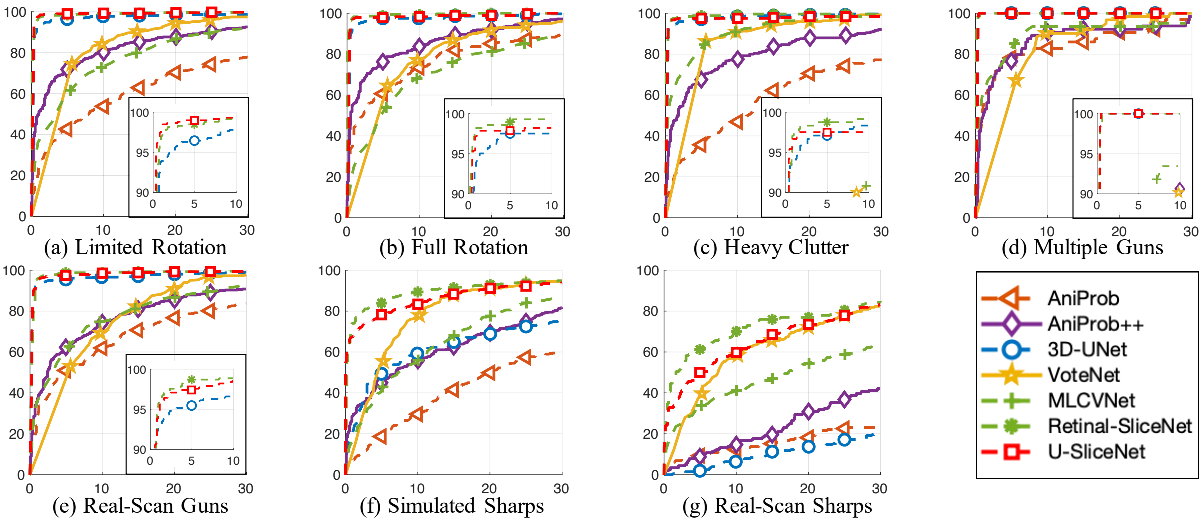
Comparison methods. We compare the proposed SliceNets with five baselines. The first baseline is 3D-UNet, which is adapted from the classic 2D FCN [23] by replacing the 2D covolution with 3D convolution. Due to the memory burden of the high dimensionality of the original input volumes, we downsample the volumes to $256 \times 256 \times 256$. During training, we further crop out small volumes of size $64 \times 64 \times 64$ from each bag, five volumes that have large overlap and five volumes that have no or small overlap with the target objects. The second baseline is Anisotropic Probing Network [31], denoted as AniProb, which uses a set of 1×1 kernels to encode the input volume to a 2D slice, and feed the 2D slice to a RetinaNet [21]. We further modify AniProb [31] to AniProb++ as a third baseline. Instead of learning a single slice, AniProb++ encodes the volume into multiple slices. We learn slices from XY, YZ, XZ directions and fuse the predicted bounding boxes from the three different directions to obtain the 3D bounding boxes. Lastly, we compare SliceNets with the state-of-the-art 3D object detection methods VoteNet [28] and MLCVNet [41], which takes the coordinates and features of points as input and outputs 3D bounding boxes. To covert our volumetric data into point clouds, we record all voxels by their (X, Y, Z) coordinates and physical densities and then remove voxels with trivial densities. During training and inference, we further subsample each point cloud to $12K$ points.

Baggage classification. We first evaluate the performance of SliceNets on 3D baggage classification, where we only seek to predict whether or not a baggage contains weapons. To perform this task using Retinal-SliceNet, we consider the largest predicted value of the 3D prediction as a bag-level score. For U-SliceNet, we use the largest bounding boxes score as the bag-level score. Figure 3 (a-g) shows the Receiver Operating Characteristic (ROC) curves on seven subsets, and Table 2 summarizes the recall for each subset at 5% false alarm rate. The proposed SliceNets outperform the state-of-the-art point cloud methods by a large margin on all datasets. Retinal-SliceNet also achieves the best classification performance of 98.71% for real guns subset and 61.27% for real sharps subset.

Table 3 compares the performances of SliceNets with different slice thickness n. Both Retinal-SliceNet and U-SliceNet work best with $n = 28$ voxels for most subsets. Note that for Retinal-SliceNet, slice with thickness $n = 40$ outperforms thinner slices on both "Full Rotation" dataset

	Name	Description	# Bags
Real Data	Clear Scan	Weapons-free bags (true negatives)	11400
	Real Scan	Bags with guns and knives. Some bags have two weapons	1001
Simulated Threat Data	Limited Angle	Base bag has low clutter and orientation of inserted weapon is limited	1600
	Full Rotation	Full range of angles in the inserted weapon	7200
	Multiple Guns	Multiple guns were inserted	686
	Heavy Clutter	Base bag has heavy electronics whose density is high	2400
	Simulated Sharps	Inserted weapon belongs to sharps -- knives, scissors, and other blades	3900

Table 1. Description of the dataset used for 3D weapons detection.



(a) Limited Rotation (b) Full Rotation (c) Heavy Clutter (d) Multiple Guns (e) Real-Scan Guns (f) Simulated Sharps (g) Real-Scan Sharps

Figure 3. **Results for 3D baggage classification.** (a-g) are ROC curves for each subset. The horizontal axis is false positive rate and the vertical axis is true positive rate.

	Ltd.	Full	Heavy	Multi	Simu.	R. S.	R. S.
	Rot.	Rot.	Clutter	Guns	Sharps	Guns	Sharps
AniProb [31]	47.39	66.67	40.42	79.69	23.64	41.42	10.78
AniProb++	71.84	76.24	67.50	76.56	44.94	62.30	8.82
3D-UNet	96.46	97.52	97.10	**100.00**	49.35	95.47	1.96
VoteNet [28]	74.49	64.54	86.25	67.21	55.47	53.02	39.71
MLCVNet [41]	61.66	53.90	84.58	84.58	42.93	62.81	33.82
Retinal-SliceNet	98.48	**98.94**	**98.75**	**100.00**	**83.90**	**98.71**	**61.27**
U-SliceNet	**98.99**	97.87	97.51	**100.00**	78.18	97.41	50.00

Table 2. **A summary of classification recall for each subset at false alarm rate** 5%**.**

and “Simulated Sharp” dataset, since thick slices capture larger fractions of rotated guns and thin blades that are more characteristic to the detector.

3D object detection. We compare the performance of the proposed SliceNets and that of AniProb++ , VoteNet and MLCVNet on 3D object detection task. We use Average Precision in 3D (AP$_{3D}$) as evaluation metrics, together with three different Intersection over Union (IoU) thresholds. Table 4 shows the detection performance for five subsets and the average performance. Note that the average AP$_{3D}$ of Retinal-SliceNet is consistently better than VoteNet and MLCVNet under different IoU thresholds. And both our SliceNets outperfom the VoteNet and MLCVNet by a large margin on Heavy Clutter subset. When cluttered objects with similar densities present in close proximity with the target objects, point cloud detection methods fail to accurately regress the bounding box locations.

3D segmentation using U-SliceNet. Lastly, we compare the proposed U-SliceNet with baseline 3D-UNet on 3D semantic segmentation task. 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 truth for each voxel is a binary label for whether being background or targets. We use Mean IoU and the accuracy for the target class as evaluation metrics. Table 5 shows that the proposed method achieves higher accuracies on all subsets, and better Mean IoU on subsets except ”Full Rotation” subset.

The performance of SliceNets on sharps subsets is worse than that on guns subsets. This can be attributed to three rea-

Slice Thickness	Ltd. Rot.	Full Rot.	Heavy Clutter	Multi Guns	Simu. Sharps	Real Guns	Real Sharps	Ltd. Rot.	Full Rot.	Heavy Clutter	Multi Guns	Simu. Sharps	Real Guns	Real Sharps
in voxels	Retinal-SliceNet							U-SliceNet						
$n = 10$	97.64	97.16	97.10	**100.00**	71.69	97.09	42.65	96.46	96.81	**97.93**	**100.00**	71.69	96.44	43.14
$n = 28$	**98.48**	98.94	**98.75**	**100.00**	83.90	**98.71**	**61.27**	**98.99**	**97.87**	97.51	**100.00**	**78.18**	**97.41**	**50.00**
$n = 40$	97.98	**99.29**	97.93	**100.00**	**84.42**	97.41	55.88	87.02	95.74	93.78	**100.00**	76.10	91.91	**50.00**

Table 3. **Results for 3D baggage Classification using different slice thickness.**

	Ltd. Rot.	Full Rot.	Heavy Clutter	Multi Guns	Simu. Sharps	Avg.	Ltd. Rot.	Full Rot.	Heavy Clutter	Multi Guns	Simu. Sharps	Avg.	Ltd. Rot.	Full Rot.	Heavy Clutter	Multi Guns	Simu. Sharps	Avg.
	AP$_{3D}$ (IoU$\geq$0.3)						AP$_{3D}$ (IoU$\geq$0.4)						AP$_{3D}$ (IoU$\geq$0.5)					
AniProb++	8.20	22.03	6.56	15.86	1.43	10.82	0.64	2.66	0.33	2.41	0.14	1.24	0.02	0.22	0.00	0.01	0.00	0.25
VoteNet [28]	80.18	90.54	69.00	70.01	50.24	71.99	66.49	**87.80**	47.42	48.28	**39.97**	57.99	49.84	70.62	30.00	34.71	**30.13**	43.06
MLCVNet [41]	80.21	89.20	66.52	64.63	47.23	69.55	64.39	81.35	50.62	49.21	34.97	56.10	47.97	57.06	33.37	29.94	25.39	38.74
Retinal-SliceNet	**91.08**	**90.55**	90.85	**66.82**	38.82	75.62	**83.25**	87.64	75.19	**58.22**	28.09	**66.48**	**68.54**	**80.72**	**56.90**	**50.99**	16.40	**54.71**
U-SliceNet	90.75	85.63	**91.93**	59.17	**58.92**	**77.28**	76.73	74.11	**76.36**	39.24	29.12	59.11	37.59	33.74	36.56	14.98	6.94	25.96

Table 4. **Results for 3D Object Detection.**

	Ltd. Rot.	Full Rot.	Heavy Clutter	Multi Guns	Simu. Sharps	Avg.	Ltd. Rot.	Full Rot.	Heavy Clutter	Multi Guns	Simu. Sharps	Avg.
	Mean IoU						Accuracy (%)					
3D-UNet	0.4913	**0.5096**	0.5052	0.4440	0.2623	0.4425	63.06	61.41	63.68	53.31	45.79	57.45
U-SliceNet	**0.5078**	0.4901	**0.5153**	**0.4511**	**0.3099**	**0.4548**	**79.73**	**80.94**	**80.96**	**76.83**	**50.60**	**73.81**

Table 5. **Results for 3D semantic segmentation.**

sons: First, as shown in Figure 4, sharps category contains a large variance of shapes including folded/unfolded knives, scissors, and some unique weapons. Second, the training samples of sharps are much fewer than that of guns. And third, the typical size and thickness of sharps is very small compared to the cluttered background, resulting in severe imbalance during detection and segmentation.

Qualitative Results. The upper two rows in Figure 4 showcase the detection performance of Retinal-SliceNet, and the lower two rows demonstrate the detection and segmentation performance of U-SliceNet for ”Real Scan” guns and sharps subsets. Note that both Retinal-SliceNet and U-SliceNet are able to detect multiple instances from the same class or from different classes in the same baggage.

Timing comparison. We compare the training and inference time complexity of SliceNets with baseline methods in Table 6. All experiments are conducted on one Nvidia TITAN Xp and Intel Xeon CPU E5-2640. Note that 3D-UNet are much more expensive to train and test compared to SliceNets. During inference, the memory footprint for Retinal-SliceNet and U-SliceNet are 5.10 Gb and 9.64 Gb.

	Training time (s)	Inference time (s)
AniProb [31]	2992	1.19
AniProb++	2992	1.66
3D-UNet	16126	35.53
VoteNet [28]	1170	6.75
MLCVNet [41]	2018	7.94
Retinal-SliceNet	1765	7.89
U-SliceNet	5386	18.87

Table 6. **Timing comparison.** For each method, we measure the training time for one epoch, the inference time and GPU memory footprint given an input volume $560 \times 560 \times 560$.

Limitations. Slice-and-fuse strategy is effective for objects that retain their distinctive shapes in their silhouettes or local projections. For objects that do not satisfy this property, 2D processing of slices would invariably lead to reduced performance. A triangle pyramid, for example, is hard to be distinguished from a triangular prism using a single 2D projection. For this reason, some of the choices we make may not be amenable for other volumetric signals. That said, our approach relies critically on having slices that have distinctive object appearances. Examples of this in-

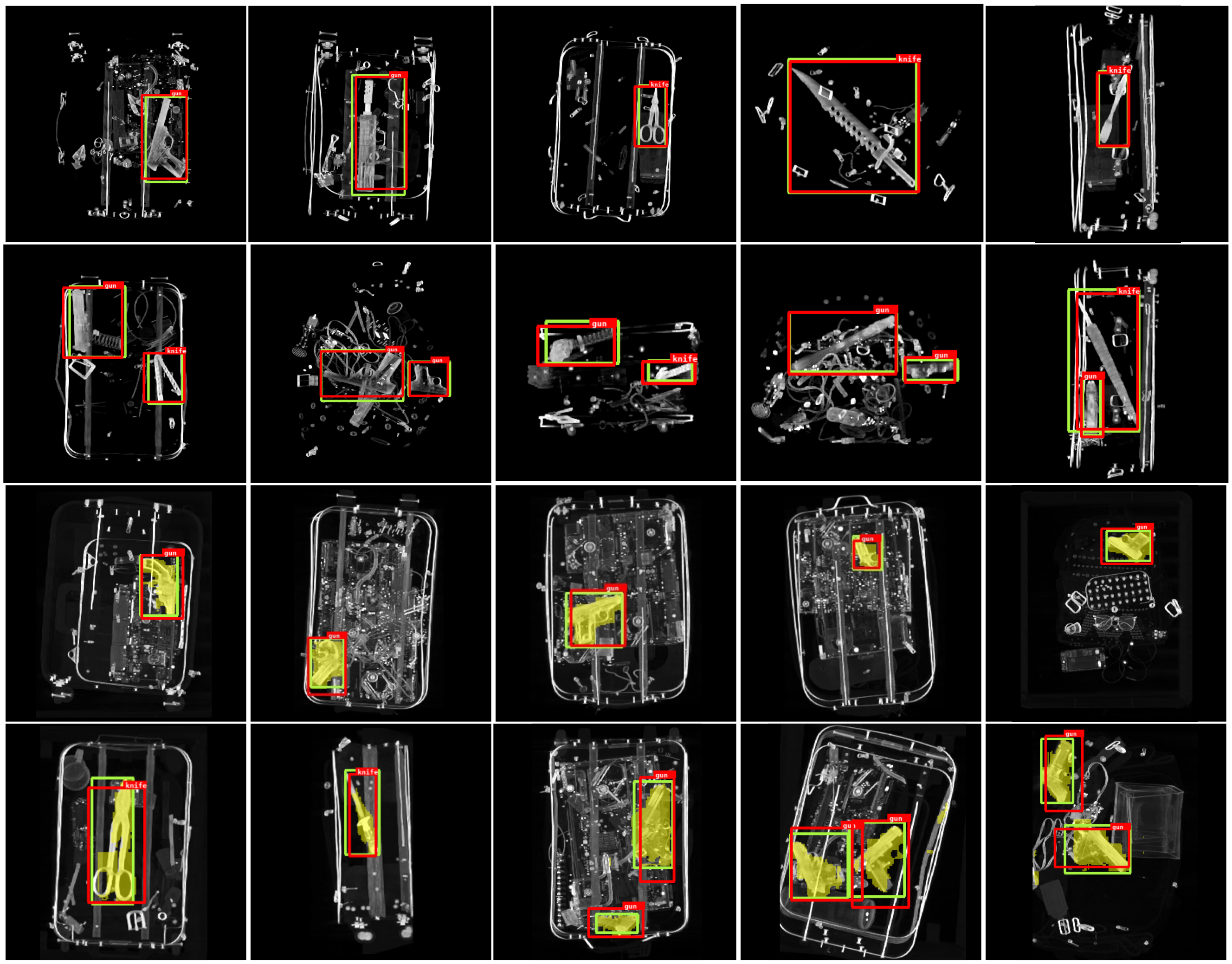


Figure 4. Qualitative results from SliceNets. The upper two rows show bounding boxes from Retinal-SliceNet and the lower two rows show bounding boxes as well as semantic segmentation masks from U-SliceNet. The red boxes are predicted bounding boxes, the yellow regions are predicted segmentation masks, and the green boxes are the ground-truth boxes.

clude CT and Magnetic Resonance Imaging (MRI) scans in medical imaging.

6. Conclusion

In this paper, we present *slice-and-fuse* strategy, a generic framework for object detection and segmentation in high-resolution 3D volumes that encodes 3D volumes into multiple 2D slices and leverages fast image-based models to obtain volumetric predictions. Based on this strategy, we further design two algorithms, called *SliceNets*, that exploit cutting-edge image-based CNNs for object detection and segmentation in 3D baggage CT scans. By training deep neural networks solely on lower-dimensional slices, our approach provides a scalable and effective way for training expressive classification and segmentation modules for high-dimensional signals.

Acnowledgement

The authors thank Ms. Bijia Chen, Mr. Mark Caron, Dr. James Connelly, and Dr. Omar AlKofahi. The authors also thank Profs. Vijaya Kumar Bhagavatula, Srinivasa Narasimhan, and David Held for insightful comments and suggestions.

This work is supported by Department of Homeland Security (DHS) Science and Technology Directorate (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 Corporation. All of the weapons images are from a DETECT 1000 that is not in a deployed system configuration.

References

[1] Simon R Arridge. Optical tomography in medical imaging. *Inverse Problems*, 15(2):R41, 1999.

[2] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In *CVPR*, 2017.

[3] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3D scans. In *CVPR*, 2018.

[4] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *ICCV*, 2017.

[5] 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 *ICRA*, 2017.

[6] Greg Flitton, Toby P Breckon, and Najla Megherbi. A 3D extension to cortex like mechanisms for 3D object class recognition. In *CVPR*, 2012.

[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] 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.

[9] Gregory T Flitton, Toby P Breckon, and Najla Megherbi Bouallagu. Object recognition using 3D SIFT in complex CT volumes. In *BMVC*, 2010.

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.

[11] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[14] Leonie Henschel, Sailesh Conjeti, Santiago Estrada, Kersten Diers, Bruce Fischl, and Martin Reuter. Fastsurfer-a fast and accurate deep learning based neuroimaging pipeline. *NeuroImage*, page 117012, 2020.

[15] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *CVPR*, 2018.

[16] Yuankai Huo, Zhoubing Xu, Yunxi Xiong, Katherine Aboud, Prasanna Parvathaneni, Shunxing Bao, Camilo Bermudez, Susan M Resnick, Laurie E Cutting, and Bennett A Landman. 3d whole brain segmentation using spatially localized atlas network tiles. *NeuroImage*, 194:105–119, 2019.

[17] 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.

[18] Bo Li. 3D fully convolutional network for vehicle detection in point cloud. In *IROS*, 2017.

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

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[24] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

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

[26] Andre Mouton, Toby P Breckon, Greg T Flitton, and Najla Megherbi. 3D object classification in baggage computed tomography imagery using randomised clustering forests. In *ICIP*, 2014.

[27] Mathias Perslev, Erik Bjørnager Dam, Akshay Pai, and Christian Igel. One network to segment them all: A general, lightweight system for accurate 3d medical image segmentation. In *MICCAI*, 2019.

[28] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019.

[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 *CVPR*, 2018.

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017.

[31] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view CNNs for object classification on 3D data. In *CVPR*, 2016.

[32] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Neurips*, 2017.

[33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, real-time object detection. In *CVPR*, 2016.

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[35] Dario Rethage, Johanna Wald, Jurgen Sturm, Nassir Navab, and Federico Tombari. Fully-convolutional point networks for large-scale point clouds. In *ECCV*, 2018.

[36] Abhijit Guha Roy, Sailesh Conjeti, Nassir Navab, Christian Wachinger, Alzheimer's Disease Neuroimaging Initiative, et al. Quicknat: A fully convolutional network for quick and accurate segmentation of neuroanatomy. *NeuroImage*, 186:713–727, 2019.

[37] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017.

[38] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. SEGCloud: Semantic segmentation of 3D point clouds. In *3DV*, 2017.

[39] Pengxiang Wu, Chao Chen, Jingru Yi, and Dimitris Metaxas. Point cloud processing via recurrent set encoding. In *AAAI*, 2019.

[40] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d ShapeNets: A deep representation for volumetric shapes. In *CVPR*, 2015.

[41] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3d object detection. In *CVPR*, 2020.

[42] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-end learning for point cloud based 3D object detection. In *CVPR*, 2018.