CSIRO at the ImageCLEFmed 2022 Tuberculosis Caverns Detection Challenge: A 2D and 3D Deep Learning Detection Network Approach

Bowen Xin[*], Hang Min[*], Ashley G Gillman, Bevan Koopman, Jason Dowling and Aaron Nicolson

[*] *Equal contribution*

Australian e-Health Research Centre, Commonwealth Scientific and Industrial Research Organisation, Herston, Queensland, 4006, Australia

Abstract

Tuberculosis (TB) is one of the leading causes of death worldwide. Automated detection of lung caverns associated with TB in Computed Tomography (CT) could help clinicians optimise treatment. However, caverns detection on 3D CT data is challenging due to the curse of dimensionality, thus requiring larger training data and more computational resource. Our team (AEHRC CSIRO) participated in ImageCLEFmed TB caverns detection 2022 to address this challenge, by developing a 2D YOLO-based model (TBdet-2D), and an efficient 3D Retina-U-Net-based model (TBdet-3D). Both networks were trained on 559 CT data with data augmentation and tested on 140 data provided by the challenge. The results show that TBdet-3D (mAP_IoU 0.504) outperformed TBdet-2D model (mAP_IoU 0.308) on testing data, indicating that employing a 3D approach instead of a 2D approach is more appropriate for the task. Our team placed first among the participating teams in this challenge. An overview of ImageCLEFmed Tuberculosis 2022 is available at: https://www.imageclef.org/2022/medical/tuberculosis.

Keywords

Tuberculosis caverns detection, Retina U-Net, YOLO, Computed Tomography

1. Introduction

Tuberculosis (TB) is one of the leading causes of death globally, leading to approximately 1.7 million deaths in 2016, according to a World Health Organisation report [1]. It is a bacteria disease caused by a germ named *Mycobacterium tuberculosis*, which can cause lung tissue inflammation and subsequently lead to lung caverns [2]. Precise detection of lung caverns in Computed Tomography (CT) imaging is an important clinical task, because it contributes to the diagnoses of the TB sub-type. This helps with optimising treatment, which is different for each TB sub-type. Even after successful treatment, the detection of lung cavern regions is of importance because the cavern may contain colonies of Mycobacterium tuberculosis

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

bowen.xin@csiro.au (B. Xin); hang.min@csiro.au (H. Min); ashley.gillman@csiro.au (A. G. Gillman); bevan.koopman@csiro.au (B. Koopman); jason.dowling@csiro.au (J. Dowling); aaron.nicolson@csiro.au (A. Nicolson)

0000-0002-4545-9574 (B. Xin); 0000-0002-9323-7167 (H. Min); 0000-0001-9130-1092 (A. G. Gillman); 0000-0001-5577-3391 (B. Koopman); 0000-0001-9349-2275 (J. Dowling); 0000-0002-7163-1809 (A. Nicolson)

that could result in unpredictable disease relapse. Detection of these lung cavern regions, is generally performed by expert radiologists through manual measurements often on 2D planes [3]; however, this manual process can be time-consuming, experience-dependant, and with limited reproducibility due to inter-observer variability. Thus, there is an urgent clinical demand to develop an automated detection algorithm for lung caverns associated with TB on CT imaging.

There have been advancements in automated object detection for natural images, mostly for 2D images. These methods can be generally categorised into two-stage and one-stage detection mechanisms. Two-stage mechanisms (e.g., R-CNN [4]) generate a sparse set of object locations in the first stage, and classify each candidate location as either foreground or background in the second stage. Algorithms along this line include fast R-CNN [5] and faster R-CNN [6]. One-stage mechanisms aim to simplify the process while retaining the accuracy through dense sampling of object scales, locations and aspect ratios. Such examples, including YOLO [7, 8], SSD [9] and RetinaNet [10], are relatively faster with matching performance to their two-stage counterparts.

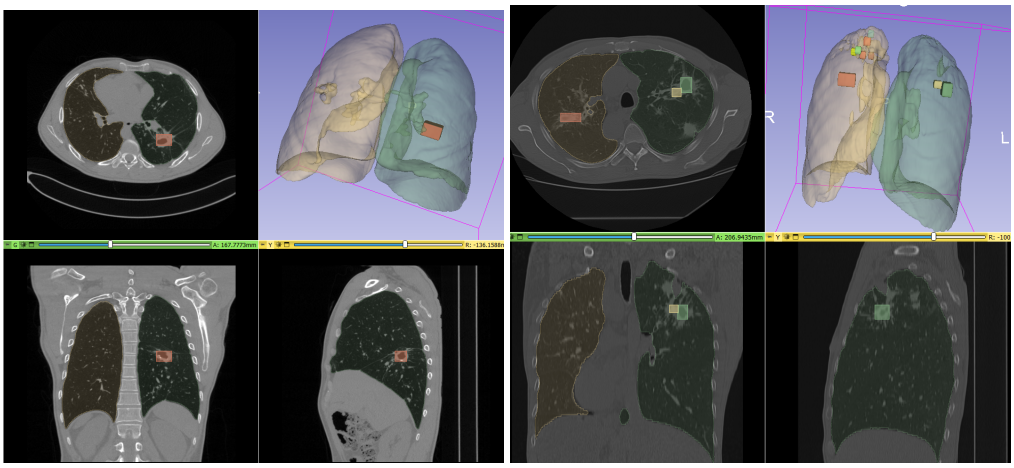
Cross-sectional imaging, such as Computed Tomography (CT), is an important tool in radiology that makes use of not only length and width, but also depth. This captures comprehensive spatial information (such as shape, geometry and location) for the detection task; however, the limitation is that processing such 3D data can be computationally expensive and data hungry due to the curse of dimensionality [11]. An alternative way to process 3D data is to operate on individual 2D slices with a 2D detection network. However, this ignores the contextual information between slices. This issue could be mitigated by considering 2D slices along different planes (e.g., sagittal and coronal planes). Furthermore, recent advancements in 3D one-stage detection networks, such as Retina U-net [12], also bring potential to achieve higher speed and accuracy at the same time.

In this work, we developed a 2D model (TBdet-2D) and a 3D model (TBdet-3D) for TB caverns detection on CT images in the imageCLEF 2022 challenge [13]. For the 2D model, YOLOv5 networks [14] were trained on axial and coronal slices of the 3D CT image. The 2D bounding boxes were then merged into 3D bounding boxes. For the 3D model, we trained a 3D Retina U-Net network [12], which leverages both the model simplicity of RetinaNet and the supervision segmentation signal from U-Net. In addition, we reduced the false positive rate by plane-based bounding box merging. Comparison of the 2D and 3D models was carried out on the TB-CT dataset provided by the Image-CLEF 2022 Caverns Detection challenge [15]. In this challenge, our 3D model achieved the 1st place with a mean averaged precision (mAP) of 0.504 (intersection over union (IoU) $\in [0.40, 0.75]$) on the testing data. Our 2D model achieved the second best mAP of 0.308.

We summarised our contributions as follows:

- We developed TBdet-2D, a 2D TB caverns region detector using YOLOv5 and an efficient 2D bounding box merging technique based on connected component labelling [16].
- We developed TBdet-3D, a 3D TB cavern region detector based on Retina U-Net, equipped with a false positive reduction module that uses plane-based bounding box (PBB) merging.
- We provide a comparison between the 2D and 3D approaches on the ImageCLEFmed TB Caverns Detection 2022 challenge.

2. Task Description



(a) TB with one cavern (b) TB with multiple caverns

Figure 1: Examples of CT tuberculosis lung caverns (from the ImageCLEFmed TB Caverns Detection challenge 2022).

2.1. Data

The dataset for ImageCLEFmed TB Caverns Detection 2022 contains 559 3D CT images for training and 140 3D CT images for testing. Each CT image has 512×512 pixels per slice and an average of approximately 100 slices, as well as its associated metadata, including the dimensions of the image, voxel spacing, etc. The raw voxel intensities for each image are stored in Housfield units. The lung masks were automatically segmented by a three-stage technique [17], and provided for each CT image by the challenge organisers. The targets in the challenge were bounding boxes of cavern regions within the CT images. The target 3D bounding boxes were extracted using manual segmentation masks delineated by radiologists. Data examples are shown in Figure 1.

2.2. Metrics

The detection task was evaluated using the mean average precision (mAP) over a series of IoU thresholds. The IoU between a predicted bounding box (predBB) and the ground truth bounding box (gtBB) is defined as:

$$IoU = \frac{predBB \cap gtBB}{predBB \cup gtBB} \tag{1}$$

For each IoU threshold t, the average precision (AP) is computed using the formula:

$$AP(t) = \frac{TP(t)}{TP(t) + FP(t) + FN(t)} \tag{2}$$

where a true positive (TP) is counted if the computed IoU is greater than the threshold t ($IoU > t$); a false positive (FP) is counted if a predBB has no associated gtBB with $IoU > t$; a false negative (FN) is counted if a gtBB has no associated predBB with $IoU > t$. The final mean average precision was defined as the mean of $AP(t)$ sweeping over a range of IoU thresholds t where $t \in (0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75)$ in this task. The evaluator was implemented in python and provided at https://github.com/SergeKo/clef2022-caverns-detection-metric.

3. Methodology

In this work, we consider two separate methodologies and associated network topologies. The first, *TBdet-2D* presented in §3.1, is based on the YOLO 2D object detection network, which frames the 3D detection task as a series of 2D detection tasks to be subsequently aggregated. The second approach, *TBdet-3D* presented in §3.2, instead is built upon an efficient 3D one-stage network, Retina U-Net.

3.1. TBdet-2D

3.1.1. YOLOv5 network

The TBdet-2D approach adopted the recent version of the YOLO [7] object detection network YOLOv5. The general concept of YOLO networks is as follows: The network applies a single convolution network on the image and divides the image into grid cells. Each grid cell predicts the object class probabilities and a number of bounding boxes. Each bounding box is represented by parameters including the confidence score of object presence, centre coordinates, height and width. Finally, the network utilises non-maximum suppression to remove overlapping bounding boxes with low confidence scores. Compared with the original YOLO structure, YOLOv5 incorporated the cross stage partial network [18] into the backbone to addresses the duplicate gradient issue, and the path aggregation network [19] as the neck for feature fusion to improve information flow and localisation accuracy. The loss function of YOLOv5 is a combination of bounding box regression loss, binary cross entropy for object presence and cross entropy for classification loss. YOLOv5 has a model ensembling feature which allows the aggregation of predictions from multiple base models to improve generalisation on unseen data. YOLOv5 has a few variants of different levels of complexity and size. For this challenge, we chose the YOLOv5s6 network, which is a small YOLOv variant with 12.6M parameters.

3.1.2. Input image pre-processing

For the TBdet-2D method, the axial and coronal slices were used for caverns detection. The pre-processing stage is illustrated in Figure 2. The 2D slices were cropped with the bounding box of the lung masks provided by the organisers. Since the YOLO network accepts 3-channel RGB images as input, we created a pseudo-RGB input which consists of the target slice in the middle and the two adjacent slices in the other two channels. The input image is linearly normalised to 8bit between 0 and 255 via min-max normalisation.

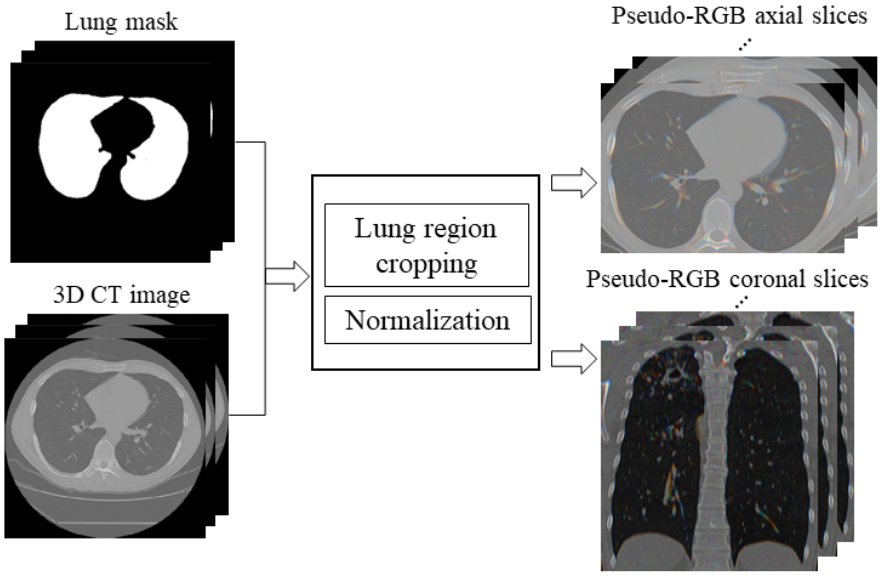


Figure 2: Pre-processing stage generating the pseudo-RGB axial and coronal slices for the TBdet-2D approach.

3.1.3. Network training

The training process of the YOLO networks is outlined in Figure 3. To train the YOLOv5 network, a 5-fold cross validation was carried out on the training data, where the cases were randomly partitioned into 5 folds with 1 fold for validation and the rest folds for training in each iteration. The YOLO networks were trained on the axial and coronal slices separately. Given the large number of axial and coronal slices in the CT images, we only selected a subset of the slices for training. The sampling strategy is as follows: for slices intersecting with the cavern ground truth bounding box, only the middle $\frac{1}{4}$ to $\frac{3}{4}$ slices were used; for slices which do not intersect with the cavern bounding box, 10 samples evenly spaced across all slices were selected. During each training iteration, the model achieving the best performance metric (a weighted sum of precision, recall, mAP ($IoU = 0.5$) and mAP ($IoU \in [0.5, 0.95]$)) on the validation set was saved. Within the 5-fold cross validation, five YOLO-axial and YOLO-coronal models were generated for caverns detection on axial and coronal slices respectively. Each YOLO network was trained for 200 epochs with a batch size of 12 and an image size of 1280×1280. Augmentations including translation, scaling, flip and mosaic augmentation [20] were applied during training. Stochastic gradient descent (SGD) was used as the optimiser with an initial learning rate (LR) of 0.01, a momentum of 0.937 and a weight decay of 0.0005. The one-cycle LR scheduler [21] was also applied with a maximum and minimum LR of 0.001 and 0.001×0.001 respectively.

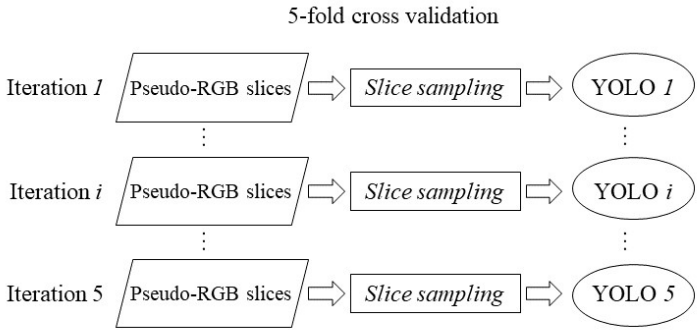


Figure 3: The training process of YOLO networks on the pseudo-RGB slices for the TBdet-2D approach.

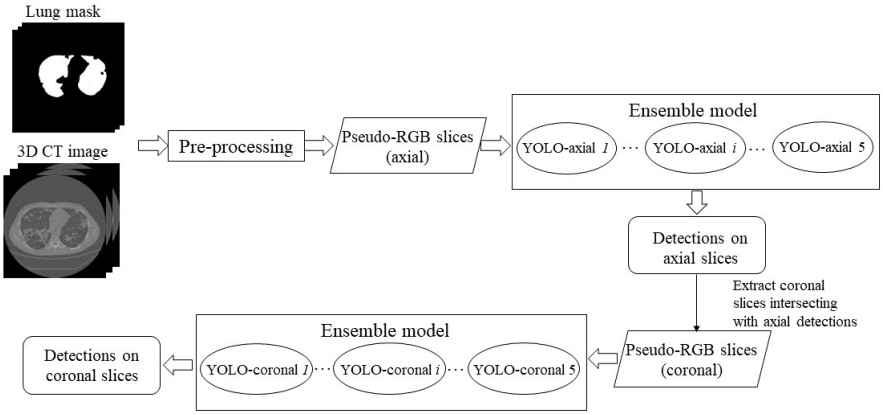
3.1.4. Cavern detection on testing data

After the YOLO networks were established on the training data, the networks were applied on the testing data to generate 2D detections as shown in Figure 4a and then the 2D detections were merged into 3D detections through connected component labelling as shown in Figure 4b.

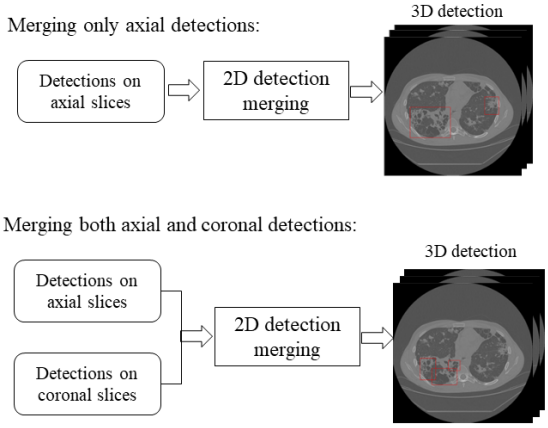
YOLO detection on testing data When using the YOLO models to perform caverns detection, the ensemble model (as described in section 3.1.1) of the 5 YOLO-axial models from the 5-fold cross validation were firstly applied onto the axial slices within the lung masks from the testing images. Then the ensemble model of the 5 YOLO-coronal models was applied onto the coronal slices that intersect with the axial detections as shown in Figure 4a. The confidence threshold for both steps was set as 0.1. The detections generated on the cropped slices (with the lung region) were finally matched back to the original full-size slices.

Merging 2D detections into 3D detections After the detections were generated on the axial and coronal slices, the 2D detections were merged into 3D ones via connected component labelling. The general merging process is as follows: 1) Given a set of 2D detections $D = \{d_n|n = 1, ..., N\}$ on slices $S = \{s_n|n = 1, ..., N\}$ with prediction probabilities of $P = \{p_n|n = 1, ..., N\}$ and a 3D binary mask M to capture the merged detections, the voxels within the bounding box represented by d_n on slice s_n were set as 1 on M if $p_n \geq Th1$ ($Th1$ is the slice probability threshold). 2) A connected component analysis was carried out on M resulting in a set of connected components $C = \{c_k|k = 1, ..., K\}$ and the bounding boxes $Bbox = \{bbox_k|k = 1, ..., K\}$ of these components. 3) If a connected region c_k consists of a subset of slices $D' \subseteq D$ which is associated with the corresponding probabilities $P' \subseteq P$, the probability for the bounding box of this connected region $bbox_k$ is set as the maximum value within P' ($P\_bbox_k = max(P')$). 4) If $P\_bbox_k \geq Th2$ ($Th2$ is the bounding box probability threshold), the bounding box $bbox_k$ is included in the final detection.

Two types of merging strategies were used as shown in Figure 4b: one is to merge only the axial detections and the other one is to merge both axial and coronal detections. For axial-only merging, the general process described above was followed. For axial-coronal merging, the axial and



(a) YOLO detection on testing data.



(b) 2D detection merging

Figure 4: Cavern detection on testing data. The YOLO ensemble models were applied on the pseudo-RGB slices to generated detections on axial and coronal slices as in subfigure (a). Then the 3D detection results were generated by either merging the detections solely on axial slices or on both axial and coronal slices.

coronal detections were firstly merged into separate 3D masks M_{axial} and $M_{coronal}$ following the step 1) and the two masks were further fused into a single mask $M = max(M_{axial}, M_{coronal})$. Then, the rest of the steps were carried out to generate the final set of bounding boxes. The slice probability threshold $Th1$ and bounding box probability threshold $Th2$ were set empirically based on the out-of-fold (OOF) validation performance within the 5-fold cross validation.

3.2. TBdet-3D

3.2.1. TBdet-3D architecture

As shown in Figure 5, the network architecture of TBdet-3D network is based on Retina U-Net [12]. Specifically, it contains two major modules, including a Unet-shaped Feature Pyramid Network (U-FPN) feature extractor and a detection head. Pre-processing is conducted before

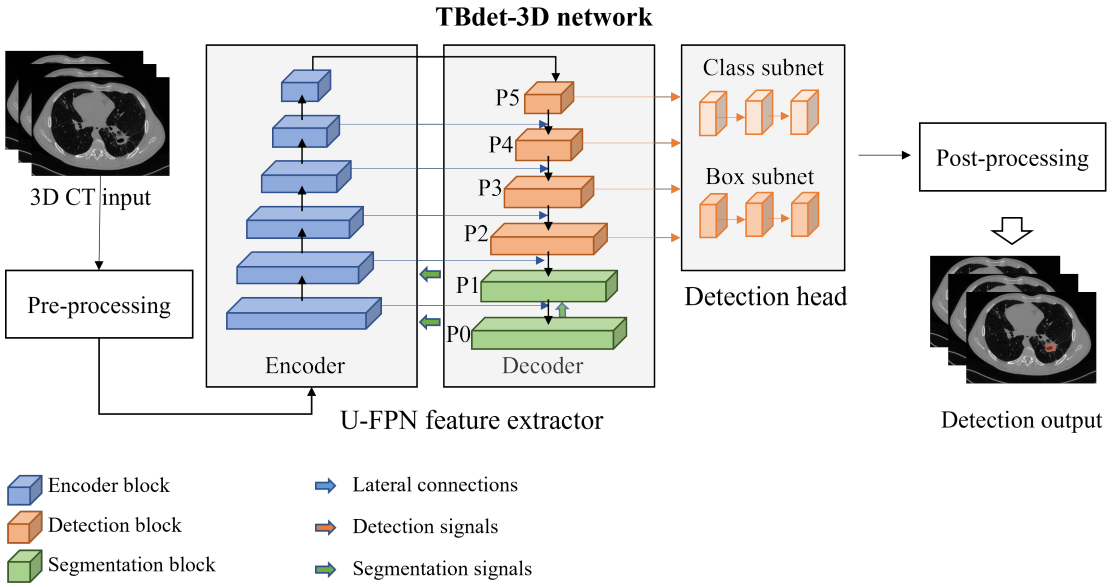


Figure 5: TBdet-3D network. The network consists of two major modules, including U-FPN feature extractor and a detection head. U-FPN is composed of a bottom-up encoder (coloured in blue) and a top-down decoder (coloured in red and green). Detection signals from P2-P5 in the decoder are fed to the detection head, while segmentation signals from P1-P2 in the decoder are fed back to the encoder. The detection head has a classification sub-network (class subnet) for label classification and a regression sub-network (box subnet) for bounding box regression.

inputting 3D CT data into the network and post-processing is implemented before the output bounding boxes.

U-FPN feature extractor U-FPN is a variant of FPN with a Unet-shape network, which is used to extract multi-scale 3D image features, while enabling semantic segmentation supervision. As shown in Figure 5, it consists of two parts, including 1) bottom-up feature encoder (coloured in blue) and 2) top-down feature decoder (in which detection blocks are coloured in red and segmentation blocks are coloured in green).

- *Bottom-up feature encoder* is a feed-forward 3D convolutional network with a hierarchy of multi-scale computation units with a scaling step of 2. In total, the encoder has six pyramid levels where each level consists of plain 3D convolution, ReLU, and instance normalisation blocks.
- *Top-down feature decoder* generates a set of higher resolution features by upsampling spatially coarser but semantically stronger feature maps using another set of 6-level pyramids with a scaling step of 2. These higher resolution features are further enhanced with features from bottom-up encoders (with the same size but more accurate localised activation) using lateral connections.
- As suggested by [12], two additional *segmentation blocks* P0 and P1 (coloured in green) are added to conventional FPN (coloured in blue and red) for the symmetry of the encoder

and the decoder. This enables fully semantic segmentation supervision.

Detection head for bounding box classification and regression The detection head is designed for one-stage dense detection. It consists of a classification sub-network for convolutional object classification, and a regression sub-network for convolutional bounding box regression. Both sub-networks use feature outputs from pyramid level 2-5 (P2-P6) from the top-down decoder as input.

- *Classification sub-network* is designed to predict the object label at each spatial position for each anchor. The classifier is a small fully convolutional network (FCN) with three convolutions with group norm, attached to P2-P6 of the decoder, with parameters shared across these pyramid levels. Along with the classifier is a regression sub-network designed to regress the offset from each anchor box to a nearby ground-truth object if it exists.
- *Regression sub-network* shares a similar network design as a small FCN except that it outputs 4 linear outputs per spatial location per anchor for prediction of relative offset between the anchor and ground-truth bounding box.
- *Anchor matching* is achieved with adaptive training sample selection [22] to deal with varying object sizes across datasets. It is not required to have the centre of anchor boxes in the ground truth box to prevent from removing positive anchors for small objects.

3.2.2. TBdet-3D loss

The TBdet-3D loss includes both detection loss and segmentation loss for better detection performance.

$$L_{TBdet-3D} = L_{det} + L_{seg} \tag{3}$$

Detection loss In the detection loss, hard negative mining is used to select 1/3 positive and 2/3 negative anchors in order to balance positive and negative anchors. Then, Binary Cross-Entropy loss is used for label classification, while Generalised IoU loss [23] is used for bounding box regression.

$$L_{det} = L_{BCE} + L_{GIoU} \tag{4}$$

Segmentation loss In the segmentation loss, Dice loss and pixel-wise cross-entropy loss are used to differentiate foreground and background pixels. A soft Dice loss, in addition to cross-entropy loss, is applied because it could stabilise training for segmentation task with class imbalance [24].

$$L_{seg} = L_{CE} - \frac{2}{|K|} \sum_{k \in K} \frac{\sum_{i \in I} u_i^k v_i^k}{\sum_{i \in I} u_i^k + \sum_{i \in I} v_i^k} \tag{5}$$

where u is the softmax output of the network (P0 logits), and v is the one-hot encoding of the ground-truth segmentation map. Both u and v share the same shape $I * K$ where I denotes the number of pixels in the training batch, and K is the number of classes.

3.2.3. TBdet-3D false positive reduction

False positive reduction is a big challenge in object detection tasks. To reduce false positives, we proposed a plane-based bounding box merging (PBB merging) to combine small bounding boxes belonging to a larger one. The motivation of this technique is that we found a considerable number of false positive bounding boxes actually corresponded to a ground truth with larger size, even after the conventional post-processing pipeline has been implemented (Non-maximum suppression (NMS), weighted box clustering [12], small-size object removal, and adjustment of prediction confidence). To address this issue, we first tried merging all overlapped bounding boxes, but it led to decreased overall performance due to significantly increased false negatives. Then, we addressed this issue by only merging bounding boxes sharing a 2D plane with a large proportion of overlapped areas.

The algorithm of PBB merging includes three steps. First, we define the connectivity of two predicted bounding boxes b1 and b2 if 1) they share an overlapping area (IoU > 0) and 2) there is a large proportion of overlap between the two boxes on any of the 2D planes (xy, yz or xz plane) with threshold 0.8. Secondly, we establish a undirected graph setting each bounding box as a node and the computed connectivity as an edge. Thirdly, we merge all connected nodes into a larger box using graph-based breadth first search. The merged box is the bounding box of all previous small boxes.

4. Results and Discussion

In this section, we summarised the 5-fold validation results of TBdet-2D and TBdet-3D in §4.1 and §4.2, respectively. In §4.3, we compared the performance between TBdet-2D and TBdet-3D, and the performance of other teams on the testing dataset.

4.1. TBdet-2D results

By choosing different 2D detection merging strategies, slice and bounding box thresholds $Th1$ and $Th2$, five different detection results were generated from the TBdet-2D approach. For the training and validation process, the OOF prediction performance on the validation sets is presented in Table 1. The YOLO-axial model with Th1=0.1 and Th2=0.8 achieved the best overall OOF prediction on the validation sets. During testing, the 5 YOLO models from the 5-fold cross validation were fused to generate predictions on the testing data. The testing results of all 5 submissions using the 2D approach are listed in Table 3. It can be observed that the ensemble YOLO-axial model with Th1=0.1 and Th2=0.8 also achieved the best mAP (0.308) among the TBdet-2D results on the testing data, which is higher than the mAP (0.295) of the runner-up team.

4.2. TBdet-3D experiments and results

4.2.1. Experimental implementation

All experiments were performed on a Linux server with Tesla P100-SXM2 GPU with 16gb RAM. Experiments were conducted using the Python language, and TBdet-3D was based on

Table 1
Out-of-fold prediction on the validation sets within 5-fold validation using TBdet-2D.

Methods	Fold 0	Fold 1	Fold 2	Fold 3	Fold4
TBdet-2D (YOLO-axial, Th1=0.1, Th2=0.8)	0.345	0.398	0.350	0.355	0.385
TBdet-2D (YOLO-axial, Th1=0.5, Th2=0.7)	0.311	0.342	0.274	0.279	0.302
TBdet-2D (YOLO-axial, Th1=0.6, Th2=0.7)	0.300	0.316	0.265	0.266	0.290
TBdet-2D (YOLO-axial&coronal, Th1=0.6, Th2=0.7)	0.282	0.328	0.244	0.261	0.349
TBdet-2D (YOLO-axial&coronal, Th1=0.5, Th2=0.7)	0.270	0.326	0.244	0.268	0.353

Pytorch-lightning and implemented with a self-configuring framework named nnDetection [25].

In the preprocessing stage, we used techniques including lung masking, resampling, normalisation and data augmentation. 1) In lung masking, we extracted the lung region of CT images using provided lung masks using a three-stage segmentation algorithm [17], while setting the remaining voxels to 0. 2) To address the heterogenous voxel spacing in CT data (especially in the z-axis), we used image resampling via third-order spline interpolation following [26]. 3) normalisation was performed based on mean and standard deviation on the training data. 4) To fit data into memory, images were cropped into patches with size [z, x, y] = [112, 160, 160], which was determined by gradually reducing the patch size until the memory constraints were fulfilled. The batch size was fixed to 4. Following [26], we applied a series of data augmentation techniques, including rotations, scaling, Gaussian noise, Gaussian bur, brightness, contrast, gamma correlation and mirroring on the fly during training.

In the training stage, the network is trained for 60 epochs with 2500 mini-batches, and half of the batch is constrained to have at least one object. For the optimisation, SGD with Nesterov momentum 0.9 is used. In the first 50 epochs, the warm-up learning rate was linear rampped up from 1e-6 to 1e-2 over 4000 iterations to reduce early overfitting, followed by polynomial decay until 50 epochs. In the last 10 epochs, we used cyclic learning rate fluctuating from 1e-3 to 1e-6 during every epoch, in which model weights were snapshot for stochastic weight averaging. 5-fold validation was performed on training data.

4.2.2. TBdet-3D main results

We investigated the TBdet-3D model with different post-processing techniques on the validation datasets. The evaluation was mAP_IoU provided by the challenge. TBdet-3D (conf_0.5) is the baseline Retina U-Net with a default prediction confidence threshold as 0.5. TBdet-3D (conf_0.7) is the Retina U-Net with prediction confidence threshold of 0.7. TBdet-3D (PBB merging) applied PBB merging technique to the TBdet-3D (conf_0.7).

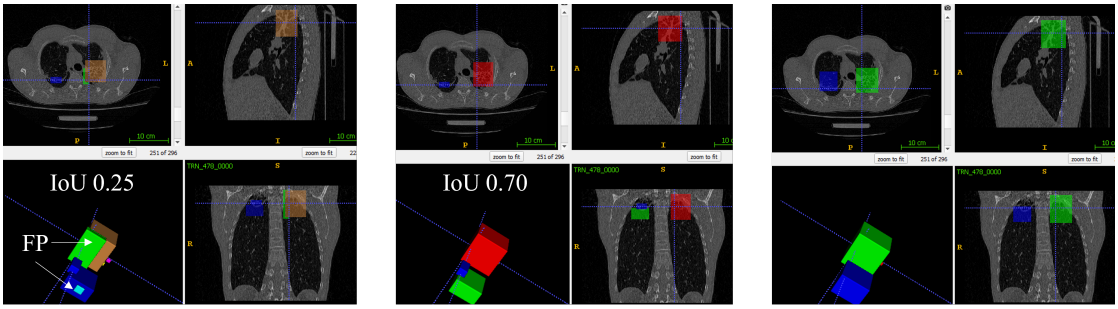
Table 2 summarises the 5-fold validation results of these three models. TBdet-3D (PBB merging) achieved the best results in all folds, followed by TBdet-3D (conf_0.7), and TBdet-3D (conf_0.5). In the results from consolidating models from 5 folds, the highest mAP_IoU acheived by TBdet-3D (PBB merging) was 0.623. The results indicate both post-processing techniques (adjustment of prediction confidence, and PBB merging) were effective.

Table 2
Results of TBdet-3D in 5-fold validation. "5 folds" column shows the results of consolidated model over all 5 folds. **Bold text** indicates best results in each column.

	Fold 0	Fold 1	Fold 2	Fold 3	Fold 4	5 folds
TBdet-3D (conf_0.5)	0.580	0.520	0.490	0.589	0.512	*0.561*
TBdet-3D (conf_0.7)	0.650	0.570	0.539	0.632	0.570	*0.609*
TBdet-3D (PBB merging)	**0.665**	**0.585**	**0.554**	**0.637**	**0.595**	***0.623***

4.2.3. Effect of PBB merging

We further investigated the effect of PBB merging as shown in Figure 6. Figure 6a and 6b show the prediction results without and with the PBB merging technique while Figure 6c shows the ground truth for this case. The IoU between the prediction and ground truth improves from 0.25 to 0.70, and it can be seen that two false positive cases closely attached to the bigger bounding boxes by plane were successfully merged into the bigger boxes.



(a) Prediction without merging (b) Prediction with merging (c) Ground truth

Figure 6: A case study comparing of TBdet-3D prediction with/without PBB merging and the ground truth.

4.3. Submission results on testing data

Table 3 shows the testing results of our model compared with other teams from the leaderboard as well as a comparison of our own submission.

Comparison of TBdet-3D and TBdet-2D in our submissions Table 3 shows that our TBdet-3D outperformed TBdet-2D model on the testing dataset. The mAP_IoU results of TBdet-3D range from 0.504 to 0.479, and the one consolidating models from all 5 folds achieved the highest performance. Comparatively, the results of TBdet-2D range from 0.308 to 0.219 with the TBdet-2D (Ensemble YOLO-axial, Th1-0.1, Th2-0.2) achieving the best performance. The improved performance of the 3D model could be due to its capacity to better exploit 3D spatial information.

Comparison with other teams in leaderboard Table 3 shows that our TBdet-3D model achieved the highest evaluation score (mAP_IoU 0.504) among all participating teams (SenticLab.UAIC: 0.295 and KDE LAB: 0.185).

Table 3
Results for submissions of ours (left) and other teams in the leaderboard (right).

Model (our submissions)	mAP_IoU	Model (leaderboard)	mAP_IoU
TBdet-3D (Consolidating 5 folds)	**0.504**	**CSIRO**	**0.504**
TBdet-3D (Consoliadting 4 folds)	0.503	SenticLab.UAIC	0.295
TBdet-3D (Best 1 fold)	0.479	KDE LAB	0.185
TBdet-2D (Ensemble YOLO-axial, Th1=0.1, Th2=0.8)	**0.308**		
TBdet-2D (Ensemble YOLO-axial, Th1=0.5, Th2=0.7)	0.281		
TBdet-2D (Ensemble YOLO-axial, Th1=0.6, Th2=0.7)	0.272		
TBdet-2D (Ensemble YOLO-axial&coronal, Th1=0.6, Th2=0.7)	0.226		
TBdet-2D (Ensemble YOLO-axial&coronal, Th1=0.5, Th2=0.7)	0.219		

5. Conclusion

In this work, we described our participation in the ImageCELFmed TB caverns detection challenge 2022. We developed and compared a 2D YOLO-based model (TBdet-2D) and a 3D Retina U-Net-based model (TBdet-3D). The results show TBdet-3D outperforms the TBdet-2D on the testing data, potentially due to the better capability of the 3D detection model in capturing 3D spatial information. We further demonstrated that postprocessing techniques such as PBB merging significantly improved the mAP_IoU score on both the validation and testing data. In future work, we aim to improve the performance by further leveraging the complementary advantages of both 2D and 3D models.

References

[1] W. H. Organization, Global tuberculosis report 2013, World Health Organization, 2013.

[2] M. A. Yoder, G. Lamichhane, W. R. Bishai, Cavitary pulmonary tuberculosis: the holy grail of disease transmission, Current science (2004) 74–81.

[3] Y. J. Jeong, K. S. Lee, W.-J. Koh, J. Han, T. S. Kim, O. J. Kwon, Nontuberculous mycobacterial pulmonary infection in immunocompetent patients: comparison of thin-section CT and histopathologic findings, Radiology 231 (2004) 880–886.

[4] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[5] R. Girshick, Fast R-CNN, in: International Conference on Computer Vision (ICCV), 2015.

[6] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).

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

[8] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.
[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single Shot MultiBox Detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.
[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
[11] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, B. Gulyás, 3D deep learning on medical images: a review, Sensors 20 (2020) 5097.
[12] P. F. Jaeger, S. A. Kohl, S. Bickelhaupt, F. Isensee, T. A. Kuder, H.-P. Schlemmer, K. H. Maier-Hein, Retina U-Net: Embarrassingly simple exploitation of segmentation supervision for medical object detection, in: Machine Learning for Health Workshop, PMLR, 2020, pp. 171–183.
[13] B. Ionescu, H. Müller, R. Péteri, J. Rückert, A. Ben Abacha, A. S. de Herrera, C. M. Friedrich, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, S. Kozlovski, Y. D. Cid, V. Kovalev, L.-D. Ştefan, M. G. Constantin, M. Dogariu, A. Popescu, J. Deshayes-Chossart, H. Schindler, J. Chamberlain, A. Campello, A. Clark, Overview of the ImageCLEF 2022: Multimedia retrieval in medical, social media and nature applications, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the 13th International Conference of the CLEF Association (CLEF 2022), LNCS Lecture Notes in Computer Science, Springer, Bologna, Italy, 2022.
[14] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, J. Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, M. T. Minh, ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, 2022. URL: https://doi.org/10.5281/zenodo.6222936. doi:10.5281/zenodo.6222936.
[15] S. Kozlovski, Y. Dicente Cid, V. Kovalev, H. Müller, Overview of ImageCLEFtuberculosis 2022 - CT-based caverns detection and report, in: CLEF2022 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, Bologna, Italy, 2022.
[16] H. Samet, M. Tamminen, Efficient component labeling of images of arbitrary dimension represented by linear bintrees, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (1988) 579–586. doi:10.1109/34.3918.
[17] Y. D. Cid, O. A. J. Del Toro, A. Depeursinge, H. Müller, Efficient and fully automatic segmentation of the lungs in CT volumes., in: VISCERAL Challenge@ ISBI, 2015, pp. 31–35.
[18] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, I.-H. Yeh, CSPNet: A new backbone that can enhance learning capability of CNN, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 390–391.
[19] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.
[20] W. Hao, S. Zhili, Improved Mosaic: Algorithms for more Complex Images, in: Journal of Physics: Conference Series, volume 1684, IOP Publishing, 2020, p. 012094.
[21] L. N. Smith, N. Topin, Super-convergence: Very fast training of neural networks using

large learning rates, in: Artificial intelligence and machine learning for multi-domain operations applications, volume 11006, International Society for Optics and Photonics, 2019, p. 1100612.

[22] S. Zhang, C. Chi, Y. Yao, Z. Lei, S. Z. Li, Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 9759–9768.

[23] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 658–666.

[24] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert, et al., nnU-Net: Self-adapting framework for U-Net-Based medical image segmentation, arXiv preprint arXiv:1809.10486 (2018).

[25] M. Baumgartner, P. F. Jäger, F. Isensee, K. H. Maier-Hein, nndetection: A self-configuring method for medical object detection, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2021, pp. 530–539.

[26] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, K. H. Maier-Hein, nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation, Nature methods 18 (2021) 203–211.