International Journal of Transdisciplinary AI
Vol. 3, No. 1 (2021) 1-24
© KS Press, Institute for Semantic Computing Foundation
DOI: 10.35708/TAI1870-126250

Object Detection and Instance Segmentation in Chest X-rays for Tuberculosis Screening

Terence Griffin[1], Yu Cao[1], Benyuan Liu[1], Maria J. Brunette[2], and Xinzi Sun[1]

[1] Department of Computer Science
University of Massachusetts, Lowell, MA, USA
{ycao,bliu}@cs.uml.edu
{terence_griffin,xinzi_sun}@student.uml.edu
[2] School of Health & Rehabilitation Sciences
College of Medicine
The Ohio State University, Columbus, OH, USA
maria.brunette@osumc.edu

Received (12/15/2020)
Revised (02/17/2021)
Accepted (02/23/2021)

Abstract. Tuberculosis (TB) is a highly contagious disease leading to the deaths of approximately 2 million people annually. TB primarily affects the lungs and is spread through the air when people cough, sneeze, or spit. Providing healthcare professionals with better information, at a faster pace, is essential for combating this disease, especially in Low and Middle Income Countries (LMICs) with resource-constrained health systems. In this paper we describe how using convolution neural networks (CNNs) with an object level annotated dataset of chest X-rays (CXRs) allows us to identify the location of pulmonary issues indicative of TB. We compare the performance of Faster R-CNN, Mask R-CNN, Cascade versions of each, and SOLOv2, demonstrating reasonable results with a small dataset. We present a method to reduce the false positive rate by comparing the location of a detected object with the known location of areas where the detected class is likely to occur in the lung. Our results show that object detection and instance segmentation of CXRs can be achieved with a dataset of high-quality, object level annotations, and could be used as part of an automated TB screening process. This work has the potential to improve the speed of TB diagnosis in LMICs, if properly integrated into the healthcare system and adapted to existing clinical workflows and local regulations.

An earlier version of the paper was presented at the Third International Conference on Transdisciplinary AI.

Keywords: Computer-aided Diagnosis; Deep Learning; Medical Imaging; Object Detection; Instance Segmentation; Tuberculosis

1 Introduction

Tuberculosis (TB), also known as the disease of the poor, is an infectious disease affecting millions of people around the world. The World Health Organization (WHO) reports that there were an estimated 10 million new cases, and 1.6 million deaths, due to TB in 2018 [1]. TB is one of the top 10 causes of death worldwide, and the leading cause from a single infectious agent [36]. TB is caused by a bacteria, *Mycobacterium tuberculosis*, which spreads through the air. When an infected person sneezes or coughs, the disease can be passed to nearby healthy individuals. When detected early the disease is treatable with a course of antibiotics for six months and the spread of the disease can be limited. When left untreated, the persistent spread of TB leads to local epidemics [36].

Efforts to reduce the burden of TB are hampered by existing health systems in Low and Middle Income Countries (LMICs) with limited healthcare infrastructure, lack of healthcare workers, and for the most part, inefficient clinical workflows. In addition, severe urban poverty and lack of political will contribute to this serious public health threat. Eight countries account for two thirds of new cases: India, China, Indonesia, the Philippines, Pakistan, Nigeria, Bangladesh, and South Africa. In the Americas, Perú has the highest rate of TB per capita [36].

A major problem contributing to delays in TB diagnosis relates to how both triage and screening processes are undertaken at local public healthcare facilities in urban poor areas often found in large LMICs cities -with populations of more than 10 million people. The length of time between first contact with a patient and the start of treatment is critical because of the risk of spreading the infection during this period. The difficulty in reading chest X-rays (CXRs) and the limited number of trained healthcare professionals has given rise to an increased interest in Computer-aided Diagnosis (CAD) systems, where computer image processing and machine learning are used to interpret these images [26,10]. The research presented here is part of a larger Community-Based Participatory Research (CBPR) project with academics and local government in Lima, Perú that aims to develop culturally-relevant systems to improve the efficiency and quality of TB diagnosis in urban poor areas [4,2,20]. One of the primary goals of this project is to increase the speed at which a TB diagnosis can be made by providing CAD tools to improve the quality and efficiency of the diagnosis process.

In this paper we investigate the use of convolutional neural networks (CNNs) to perform object detection or instance segmentation of several common pulmonary issues indicative of TB in CXRs. We refer to these issues as "TB manifestations" in the remainder of this paper. Our rationale for this work is that providing the diagnosing physician with the location of TB manifestations in a CXR can lead to an improvement in the speed of diagnosis.

We chose to use object detection or instance segmentation over whole image classification because of the increased usefulness of more fine-grained information. Object level classification also lessens the problem of explainability associated with the black box nature of deep learning approaches, where a system generates an answer but lacks a trail of logic or evidence to support the result. Providing the location of the problem area in the CXR should help make the results of the system more transparent.

Object detection involves identifying objects in an image and providing their locations as bounding boxes. Instance segmentation provides the location of an object as a pixel-wise mask, rather than a bounding box. Instance segmentation is a more difficult task but provides better information. We compare the performance of a region-based CNN (R-CNN) [7] model for object detection, with a Mask R-CNN [13] model for instance segmentation. For our goal of providing a physician with information leading to fast and accurate evaluation of a CXR, we would prefer to use instance segmentation, because a pixel-wise mask provides more detailed information than a bounding box. This choice is reasonable only if the instance segmentation task can be accomplished with at or near the same performance as the object detection task.

Overall, the R-CNN model shows slightly better performance than the Mask R-CNN model, with both architectures showing similar performance at an intersection over union (IoU) threshold of 0.5. We also investigate using the Cascade R-CNN approach presented in [3], which is reported to improve the performance of both R-CNN and Mask R-CNN models. We did not see any increase in performance from this approach on our dataset.

We also compare the performance of our Mask R-CNN model with a simpler single-stage SOLOv2 [35] model. For our task the more complex two-stage Mask R-CNN model showed superior performance.

The networks developed here have a strong bias toward false positives over false negatives, as we would prefer to misclassify a CXR from a healthy patient over an infected patient. We are able to reduce some of these false positives by using domain specific knowledge. Some of the manifestations we are interested in occur only in certain locations in a lung. We show a novel method for using this information to reduce the number of false positives without any corresponding increase in false negatives.

The dataset we use is comprised of 1,186 CXR images with 2,493 instances annotated with individual masks by an expert pulmonologist. Four manifestations of TB were identified with sufficient occurrences for training a deep learning model: Airspace Consolidation, Cavitation, Lymphadenopathy, and Pleural Effusion. Eighty percent of the dataset was used for training, ten percent for validation and tuning of the hyperparameters, and ten percent for final testing.

An ensemble approach was taken, with separate networks trained for each manifestation. The results of this experiment show that our Mask R-CNN models have an average Area Under the Receiver Operating Characteristic Curve (AUC) value of 0.733 for the four manifestations. With the detection threshold set at 0.50 the networks show an average precision of 0.501 and the average recall of 0.709. A threshold of 0.50 was chosen as a reasonable default value for analysis. Adjusting this value will shift the bias between false positives and false negatives.

Manual review of the performance on the test set also supports our conclusion that the networks are able to learn the location of the manifestations.

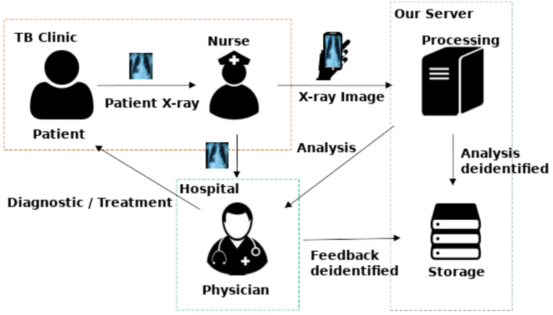


Fig. 1. eRx workflow

2 Background

Often one of the first steps in diagnosing a patient with symptoms of lung disease is to request a CXR, together with a physical examination and a sputum test. After receiving the lab results, the time taken to evaluate the CXR and decide on a course of treatment is an essential factor in treating TB and preventing the further spread of the disease.

The eRx system [4,2] is a multi-faceted CAD system developed for our larger project, combining mobile computing, cloud computing, and machine learning to improve the diagnosis of TB. The technical contribution of our project is two-fold: to provide an accurate and prompt TB screening and diagnosis tool, and to evaluate the use of CNNs to provide automated analysis of CXRs. A relevant aspect of our project includes the availability of a cloud-based system to improve the screening process by allowing TB nurses to capture and upload a CXR to a server using a cell phone. One or more physicians (who are usually in other healthcare establishments) are then notified that a CXR has arrived in the system and is available for review. The physician(s) can then respond to the nurse in a timely manner. In addition to the physicians' diagnoses, we evaluate the CXR using two CNNs, one which classified the CXR as normal or abnormal, and one which determined the most prominent manifestation of TB. A preliminary version of the system has been implemented and was recently used in a pilot study in Lima, Perú [32].

The workflow of the system (see Fig. 1) is as follows:

1. A patient visits a TB clinic and a referral for a CXR is given.
2. When the patient returns to the TB clinic, the nurse uses the eRx mobile phone app to take a picture of the CXR and upload it to a cloud server.
3. On the eRx server, two CNNs analyze the image to identify manifestations of TB.

4. One or more assigned physicians are notified via email and text message that a new CXR is available for review.
5. The physician evaluates the image using the eRx application (web or mobile) and provides immediate feedback (via text) on next steps to the nurse.
6. The physician evaluates the results of the neural networks.

The physician was not given the results of the neural networks prior to performing their own diagnosis. After completing the diagnosis the physician was asked to evaluate the performance of the neural networks and provide feedback on the correctness and usefulness of the CNN results. At this stage of the project one of the goals was to determine how well the CNNs perform compared to the physicians and to collect data on any deficiencies.

The first CNN (normal/abnormal) tells the physician whether or not the system detects a problem that could be indicative of TB. The system is essentially telling the physician “This image looks like it has a problem”. The obvious response from the physician is then “What is the problem?”. The second network, which identifies the most likely manifestation, attempts to answer this question, for example telling the physician “There is some cavitation”. The next likely question from the physician is then “Where in the image is cavitation detected?”. The work presented here investigates the possibility of providing that answer.

To build models for object detection or instance segmentation on a CXR we first need an annotated dataset. Unfortunately, there is a lack of a high quality, large dataset containing object level segmentation of manifestations associated with TB. Therefore, one aspect of the larger joint project is the construction of an annotation application to provide this data. The result is a Web application, described in full in [2], which allows an expert to graphically identify areas of interest in a CXR and provide the annotation, including the manifestation, location, confidence level, and additional notes. This data was used to train the CNNs described here.

3 Related Work

Over the past decade, CNNs have emerged as the tool of choice for solving image analysis problems [8]. This has been demonstrated by the ever increasing performance on tasks such as the ImageNet [6] and COCO [19] challenges. Recently, deep learning methods have been used successfully for medical image analysis, outperforming other methods [30,15,27,33].

The use of CNNs for CAD applied to TB diagnosis has had some success. An analysis of the performance of different CNN architectures on two small publicly available datasets is given in [24]. This work demonstrates that modern CNN approaches outperform previous methods on these datasets.

The advantage of deep learning over previous image processing methods is also highlighted in [14], where whole image classification was applied to TB screening using three datasets, reporting an AUC score of 0.96. [11] provides a review of studies applying AI to CXR analysis targeted at TB, and shows the recent prevalence of deep learning methods. Previous research on the eRx system [4,2,20] also demonstrated that, given a properly annotated dataset, an approach using deep learning provides superior results for the identification of TB manifestations in CXRs over traditional image processing methods.

Deep learning requires large, annotated datasets. For natural images this problem has been successfully addressed using crowd sourcing [6,19]. However, this approach is not appropriate for medical images, where the annotation must be performed by an expert or inferred from supporting documentation. Two large annotated datasets for CXRs are available: the ChestX-ray14 [33] dataset contains 112,120 images from 32,717 patients and the CheXpert [27] dataset contains 224,316 images from 65,240 patients. The annotations for both of these datasets were created by text mining the associated radiological reports. For our purposes these datasets are inadequate both because the annotations include symptoms of many pulmonary diseases but few instances of TB manifestations, and because of the lack or sparsity of object level annotations. The CheXpert dataset contains whole image annotations only, while the ChestX-ray14 dataset contains bounding box annotations for less than 200 instances of each pathology. Existing datasets specifically for TB are much smaller and lack object level annotations. These include the Montgomery County dataset with 138 images and the Shenzen Hospital dataset with 662 images [16,17].

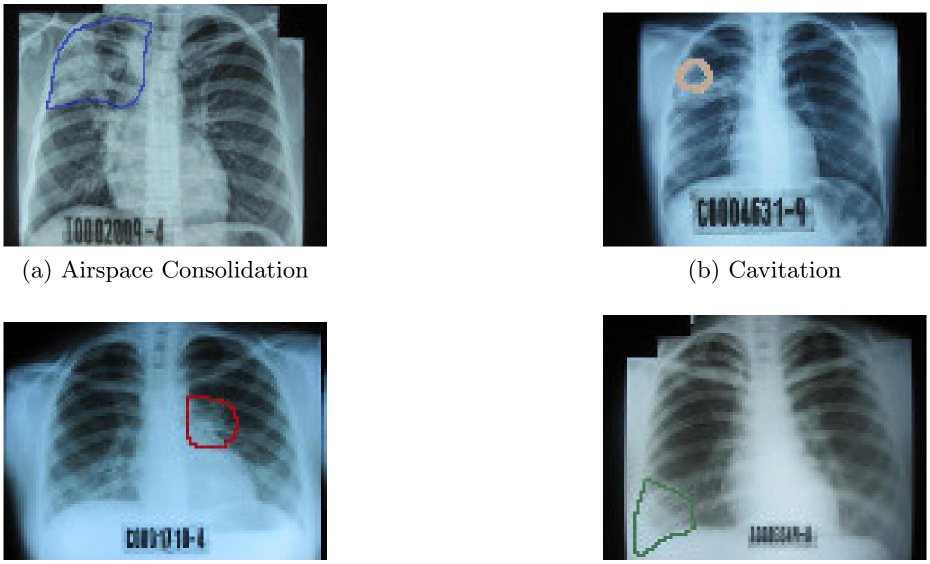
Recently, two new datasets have become available. ChestX-Det10 [21] consists of 3,500 CXRs from the ChestX-ray14 dataset with bounding box annotations of 10 pathologies. TBX11K [22] consists of 11,200 images with whole image labels for four classes: Healthy, Active TB, Latent TB, and Sick Non-TB. Additionally, there are bounding box annotations for the Active TB and Latent TB instances, with 924 and 212 images respectively. These datasets may prove useful for advancing the state of the art for the object detection task on CXRs, however they lack annotations needed for instance segmentation and identification of specific abnormalities related to TB.

For these reasons we chose to construct our own dataset with object level annotations of TB manifestations. To our knowledge this is the first effort focused on object detection or instance segmentation of CXRs to aid in the automated screening or diagnosis of TB.

4 Approach

4.1 Dataset

Our dataset consists of 1,186 images annotated using the eRx annotation system by a pulmonologist who is an expert in TB diagnosis. The images come from a set of 4,701 images provided to us by Partners in Health Perú, a non-profit organization based in Lima. The annotation system supports eleven common TB manifestations. Of these there are four for which we have over 100 instances, which may be sufficient to reliably train a model: Airspace Consolidation, Cavitation, Lymphadenopathy, and Pleural Effusion (see Fig. 2). Therefore, for the work presented here, we restrict the models to the identification of these four classes. Table 1 shows the distribution of each manifestation across the image set.



(a) Airspace Consolidation

(b) Cavitation

(c) Lymphadenopathy

(d) Pleural Effusion

Fig. 2. Example images

Table 1. Dataset Sizes

Manifestation	**Images**	**Instances**
Airspace Consolidation	1,112	1,545
Cavitation	338	441
Lymphadenopathy	327	367
Pleural Effusion	136	140

The cost in terms of time and expertise for creating this dataset is high in comparison with the cost of annotating natural images. As mentioned earlier, crowd sourcing is not an option, since we need subject matter experts to competently perform the annotations. This is a tedious task requiring a significant amount of time from highly trained professionals, and we would like to have some confidence that this is a worth while use of their time. One important result of this work is demonstrating that the effort in annotating these images can indeed lead to better networks, and ultimately better and faster diagnoses.

4.2 Networks

A Region-based CNN (R-CNN) [7] network predicts bounding boxes and classes for objects in an image. In our case the objects being detected are the TB manifestations identified by the pulmonologist. Faster R-CNN [28], shown in Fig 3, is an efficient architecture for this task which has demonstrated state of the art performance.

The input CXR is first passed through a number of convolutional layers to create a set of feature maps. This is known as the backbone network. The structure of the convolutional layers used here is typically taken from networks

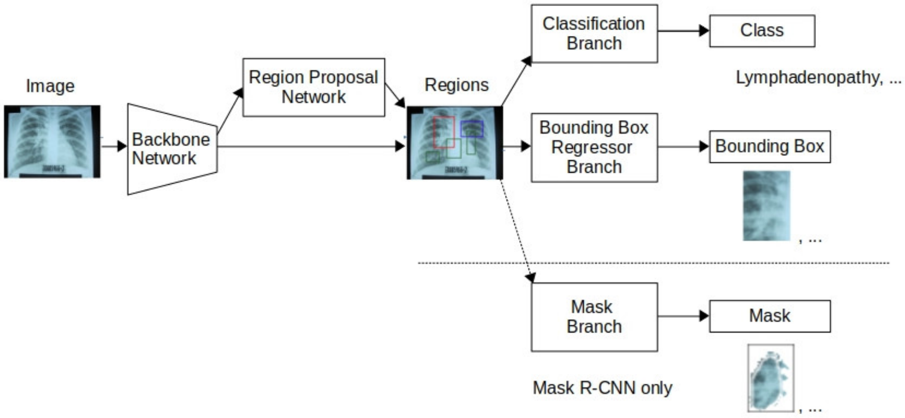


Fig. 3. Faster R-CNN and Mask R-CNN structure

developed for the object detection task, and these layers are often pretrained on a large dataset of natural images, such as ImageNet or COCO. The responsibility of the backbone network is to learn how to extract features from the input image which can be used to identity and classify objects in the image.

The feature maps are used as input into a Region Proposal Network (RPN), a small fully convolutional network [23] which generates a list of Regions of Interest (ROIs) containing objects. A sliding window approach is used to create a collection of candidate ROIs using a fixed set of sizes and scales. The RPN learns how to select a relatively small number of the best ROIs from these candidates.

The final portion of the network contains two branches, each containing a small fully connected network: the object classification branch, which predicts the class of the object, and the bounding box regressor branch, which fine tunes the size and location of the bonding box for the object within the ROI. The input to both branches is created from the feature maps produced by the backbone network in the region proposed by the RPN. The two branches are trained in parallel, which provides a simpler network than multi-stage detection schemes [28]. The network is trained using a multi-task loss, combining the losses from the classification and bounding box detection branches.

Faster R-CNN is a two-stage network, with the RPN forming the first stage and the classification and bounding box regressor networks, known collectively as the network head, forming the second stage.

Mask R-CNN [13] expands on the Faster R-CNN architecture by adding an additional branch to identify a mask for the object (shown in Fig. 3), providing more detailed location information. This additional branch produces a loss term for the location of the mask, which is combined with the losses defined by the base Faster R-CNN network during training. Like Faster R-CNN, Mask R-RCNN is a two-stage model, using the same backbone, RPN, and head network architecture. The significant differences between the two networks is the addition of the mask branch and some changes to the way the features within an ROI are pooled to improve the accuracy of the resulting mask.

The implementation of Mask R-CNN contains outputs for both the bounding box branch and mask branch of the network. This allows us to compare the performance of the simpler bounding box task to the mask task for the same network. However, because the loss values from the mask branch are used during training, the performance of the network on the bounding box task may be worse than that of a similar R-CNN network. An R-CNN network may outperform a Mask R-CNN network on the bounding box task because it is focused solely on that task, whereas the Mask R-CNN network is influenced by both tasks. To assess this difference we provide results for both the object detection task using Faster R-CNN and the instance segmentation task using Mask R-CNN.

One problem identified with Faster R-CNN based models (including Mask R-CNN) is that the model is trained using a single IoU threshold value to discriminate positive and negative samples. A low threshold leads to noisy detections and a high threshold can degrade performance [3]. The Cascade R-CNN [3] architecture seeks to reduce this issue by using a sequence of detectors trained with increasing IoU thresholds. In the implementation used for this work these detection heads are trained using IoU thresholds of 0.5, 0.6 and 0.7. We show results below for models using the Cascade and original versions of both Faster R-CNN and Mask R-CNN.

Recently, significant advances have been made in single-stage network architectures for instance segmentation. SOLOv2 [34,35] is one leading example of a network design which is simpler than Mask R-CNN while reporting similar performance. Fig 4 shows the basic structure of the SOLOv2 network. Similar to Faster/Mask R-CNN, a backbone network is used to create a set of feature maps from the input image.

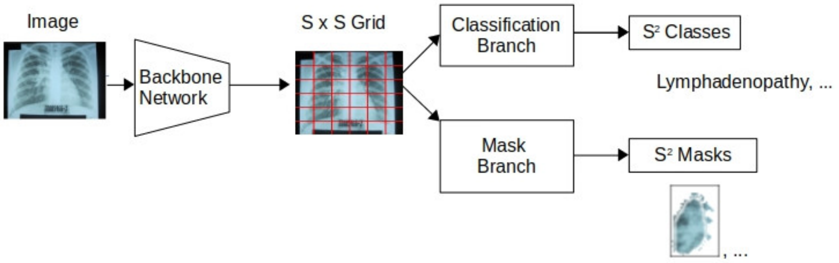


Fig. 4. SOLOv2 structure

Rather than using an RPN to propose object locations, the image area is then divided into an $S \times S$ grid. If the center of an object falls within a grid cell, then that cell is used to predict the class and mask for the object (the network will identify at most one object at each grid location). A classification branch generates class scores for each cell. The mask branch determines which pixels in the feature map belong to the object, creating S^2 masks of the full size of the feature map from the backbone network. If the classification branch determines that a cell contains an object (i.e., the score for the highest class is above the detection threshold) then the class and mask for that cell is produced as an output of the network. Similar to Faster/Mask R-CNN, the classification and mask branches are trained in parallel.

In this work we investigate both Mask R-CNN and SOLOv2 to determine if the simpler single-stage approach can achieve that same level of performance as Mask R-CNN given our dataset and task. Identifying abnormalities in CXRs is a more difficult task than identifying objects in natural images, at least for humans, requiring specialized training. However, as our input images are in some ways very similar, all being CXRs and differing more in details than in overall structure, a simpler network may be able to learn the pertinent details better than it can deal with the large variety possible in natural images. On the other hand, learning fine details may be require a more complex network structure.

4.3 Using the Object Location for False Positive Reduction

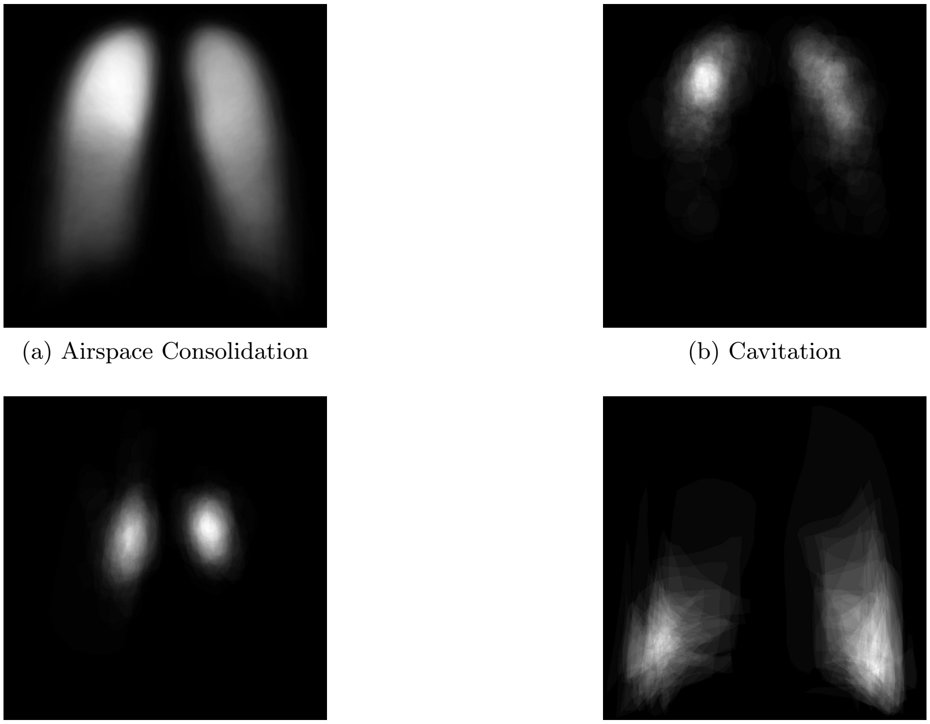
One property of CNNs is that they are translation invariant [8], meaning they are able to find an object regardless of the location in the image. While this is a useful property in general, for the task of abnormality detection in a CXR, it can lead to unnecessary false positives.

Some manifestations occur in only certain locations within the lungs. In particular, Pleural Effusion only appears at the bottom of the lungs, and Lymphadenopathy only appears near the lymph nodes, which are located to the left and right of the spine. We can use this information to filter out some false positives based on the location of the detected area.

To encode the likeliness of a manifestation occurring at a particular location we created heatmaps based on the training and validation datasets for each manifestation. A 1,024 x 1,024 pixel heatmap was constructed from the annotation data by counting the number of instances whose mask includes each pixel. The values were then normalized to a range of 0-1.

The resulting heatmaps are shown in Fig. 5. The images are reasonable given the nature of each manifestation, and in particular show the correct location for both Pleural Effusion and Lymphadenopathy. The very limited number of samples for Pleural Effusion leads to the roughness of the heatmap, where distinct polygons can be seen. The heatmap for Airspace Consolidation looks smoother, due to the large number of instances included in the dataset. The heatmap for Cavitation shows a hot spot on the right lung and does not fill the lung area completely. It is unclear if these are true properties of the manifestation or are due to the limited size of the dataset.

The per manifestation heatmaps are used during the evaluation process for each network, as a post-processing step after running the model on the input. For each instance detected by the model, the bounding box or mask is compared with the heatmap for the identified manifestation. If the average value in the area of the heatmap covered by the bounding box or mask is below a certain minimum threshold then the instance is deemed a false positive and removed from the


(a) Airspace Consolidation

(b) Cavitation

(c) Lymphadenopathy

(d) Pleural Effusion

Fig. 5. Heatmaps generated from the training and validation sets

set of detected objects. The threshold value of 0.18 was arrived at empirically through testing against the validation sets. This evaluation is performed against the test set after training and on the validation set during training.

4.4 Experiment Setup

For each network architecture evaluated, an ensemble of four separate networks were trained, one for each manifestation. The dataset used for each manifestation consists of the images annotated with that manifestation plus an equal number of images which do not contain that manifestation. Each dataset was divided into train, validation, and test sets using an 80-10-10 split. The validation set was used to tune the hyperparameters during development. The test set was used for the final evaluation of the models and was not used during training. The same dataset was used for each network architecture evaluated.

The networks were implemented using PyTorch[3] [25] and the Detectron2[4] [37] and MMDetection[5] [5] libraries. Each model was trained for a maximum of 8,000 epochs with early stopping, using a learning rate starting at 0.005 and a

[3] https://pytorch.org/

[4] https://ai.facebook.com/tools/detectron2/

[5] https://github.com/open-mmlab/mmdetection

weight decay of 0.0001. The backbone networks were initialized using pretrained weights from the ImageNet dataset, as this provided faster convergence without a significant loss in performance. The networks were trained on a Linux server with 64 GB of memory and two nVidia GTX 1080 Ti GPUs, each with 11 GB of memory. Training of each model takes between three and five hours, depending on the network architecture.

5 Results

5.1 Backbone Network Selection

The architecture of the backbone network used for feature extraction is separable from the overall structure of Faster R-CNN, Mask-RCNN, or SOLOv2 and can be easily changed. For this work we evaluated AlexNet [18], GoogLeNet [31], VGG [29], ResNet [12], and ResNeXt [38] with our Mask R-CNN model. A comparison of the performance is shown in Table 2 using the COCO evaluation metrics [19] of mean average precision (mAP) and average precision at an IoU threshold value of 0.5 (AP50). We observe that ResNet and ResNeXt outperform AlexNet, GoogLeNet, and VGG, due to the more complicated structure of the networks. Based on these results, we restricted further experiments to use either ResNet or ResNeXt backbone networks.

Table 2. Backbone Comparison for Mask R-CNN

Network	**Depth**	**mAP**	**AP50**
AlexNet	8	0.2019	0.4621
GoogLeNet	22	0.2123	0.4902
VGG	19	0.2397	0.5327
ResNet-50	50	0.2461	0.5744
ResNet-101	101	0.2610	0.5868
ResNet-152	152	0.2361	0.5330
ResNeXt-50	50	0.2768	0.6798
ResNeXt-101	101	0.2964	0.6736

5.2 COCO Evaluation Metrics

To compare the performance of the different network architectures we begin by using the COCO evaluation metrics mAP and AP50, shown in Tables 3 and 4, respectively (AC = Airspace Consolidation, CA = Cavitation, LY = Lymphadenopathy, PE = Pleural Effusion).

The performance between the four manifestations varies widely. This is due at least in part to the difference in the number of samples and the average size of each manifestation. Airspace Consolidation has both the largest number of instances and largest instances, although the value shown here is artificially high as explained in the following section. Pleural Effusion has the fewest number of instances and each instance tends to be relatively small. There may also be

Table 3. COCO mAP Comparison

Network	**Backbone**	**AC**	**CA**	**LY**	**PE**
Faster R-CNN	ResNet-50	0.3890	0.0870	0.0760	0.0391
Faster R-CNN	ResNet-101	0.4006	0.1153	0.0918	0.0629
Faster R-CNN	ResNet-152	0.3585	0.0723	0.1462	0.0315
Faster R-CNN	ResNeXt-50	0.3925	0.1102	0.1398	0.0530
Faster R-CNN	ResNeXt-101	0.4218	0.1527	0.1628	0.0893
Mask R-CNN	ResNet-50	0.3730	0.0799	0.0725	0.0390
Mask R-CNN	ResNet-101	0.3912	0.0978	0.0783	0.0419
Mask R-CNN	ResNet-152	0.3618	0.0712	0.0638	0.0325
Mask R-CNN	ResNeXt-50	0.4001	0.1065	0.1274	0.0518
Mask R-CNN	ResNeXt-101	0.4123	0.1492	0.1418	0.0864
Cascade Faster R-CNN	ResNeXt-101	0.4223	0.1523	0.1621	0.0884
Cascade Mask R-CNN	ResNeXt-101	0.4111	0.1502	0.1399	0.0865
SOLOv2	ResNeXt-50	0.3358	0.0800	0.0653	0.0201
SOLOv2	ResNeXt-101	0.3512	0.0838	0.0775	0.0261

Table 4. COCO AP50 Comparison

Network	**Backbone**	**AC**	**CA**	**LY**	**PE**
Faster R-CNN	ResNet-50	0.7789	0.2805	0.3276	0.2298
Faster R-CNN	ResNet-101	0.8002	0.2886	0.3387	0.2813
Faster R-CNN	ResNet-152	0.7125	0.2678	0.3159	0.1807
Faster R-CNN	ResNeXt-50	0.7918	0.2700	0.3412	0.2612
Faster R-CNN	ResNeXt-101	0.8607	0.3991	0.4700	0.4095
Mask R-CNN	ResNet-50	0.7830	0.2800	0.3125	0.2305
Mask R-CNN	ResNet-101	0.7881	0.2902	0.3404	0.2711
Mask R-CNN	ResNet-152	0.7049	0.2801	0.3356	0.2310
Mask R-CNN	ResNeXt-50	0.7880	0.2852	0.3254	0.2518
Mask R-CNN	ResNeXt-101	0.8609	0.3992	0.4667	0.4086
Cascade Faster R-CNN	ResNeXt-101	0.8598	0.3897	0.4698	0.4100
Cascade Mask R-CNN	ResNeXt-101	0.8607	0.3902	0.4701	0.4112
SOLOv2	ResNeXt-50	0.7485	0.2054	0.2784	0.2111
SOLOv2	ResNeXt-101	0.7719	0.2141	0.2918	0.2302

inherent characteristics of some manifestations which make them more difficult to detect than others.

Within each manifestation the comparison between different network architectures is similar. Faster R-CNN has slightly better performance than Mask R-CNN on mAP and little difference on AP50. This makes some sense due to the simpler task, which allows for more correct detections at higher IoU thresholds. Both networks achieved the best performance using the ResNeXt-101 backbone network. The larger ResNet-152 network showed a significant drop off in performance relative to the ResNet-101 and ResNet-50 networks.

The Cascade versions of the best performing Faster R-CNN and Mask R-CNN networks did not show a significant improvement over the base networks in these tests. One possible reason that we did not see similar performance gains to those presented in [3] is the limited number of samples in our dataset. The advantage of multiple detectors using different IoU thresholds may not be apparent without a larger or more varied dataset.

The single-stage SOLOv2 network did not perform as well as the two-stage networks. The small size of many of the instances in our dataset may be a factor in this gap. The results from [35] show that while SOLOv2 may outperform Mask R-CNN on medium and large size objects, Mask R-CNN performs better on small objects. It may also be that the small size of our dataset makes it difficult for the simpler network to learn as well as the more complex two-stage approach.

Although the tasks and datasets are not same, a useful comparison can be made between the performance presented here and the results shown in [21] and [22] for object detection using the ChestX-Det10 and TBX11K datasets, respectively. The ChestX-Det10 annotations contain bounding box information for 10 pulmonary diseases or abnormalities for 3,543 CXRs from the NIH ChestX-ray14 [33] dataset. Table 5 shows the results presented in [21] for a Faster R-CNN network. The average AP50 score across the ten classes is 0.450, which is in the range of our results for the Cavitation, Lymphadenopathy, and Pleural Effusion from Table 4.

Table 5. ChextX-Det10 AP50 Results

Atelectasis	Calcification	Consolidation	Effusion	Emphysema
0.364	0.516	0.599	0.502	0.666
Fibrosis	Fracture	Mass	Nodule	Pneumothorax
0.439	0.419	0.410	0.248	0.335

The TBX11K dataset contains 1,136 CXRs for which bounding box annotations are provided for two classes: Active TB and Latent TB, with 924 and 212 instances, respectively. [22] shows results for object detection giving a best AP50 score of 0.670 for the Active TB class and 0.099 for the Latent TB class. Our results (and those from [21]) fall between these two extremes. The marked difference in scores for the two classes is likely due to the class imbalance, which can also be seen in our results.

Note that the results in Tables 3 and 4 cannot be compared with the results posted on the COCO competition leaderboard[6] or with published results showing state of the art performance on the COCO dataset, due to the wide differences in dataset sizes, the type of image (natural versus CXR), and the task difficulty. The results here, however, are sufficient to conclude that the approach has promise for our application.

5.3 Manual Evaluation of Mask R-CNN

Our dataset has some peculiarities that make the usual COCO evaluation metrics less than ideal. There are two major problems. First, there is often no hard boundary between a manifestation and the surrounding tissue, and therefore the ground truth segmentation may not be drawn consistently tight. This can lead to the intersection over union (IoU) measure between the ground truth and detection to be less than would be expected for natural images.

The second issue is that there are many cases where the ground truth consists of two or more adjacent areas and the network merges these together or, conversely, a large ground truth area is detected as two separate areas, as in Fig. 6 and 7. These figures, and similar ones that follow, show the ground truth segmentation on the left and the detected segmentation on the right. In Fig. 6 the detected regions roughly correspond to the union of the ground truth regions, and in Fig. 7 the union of the detected areas covers the annotated ground truth area. Both of these examples should be considered successful but may be missed by using the standard COCO metrics.

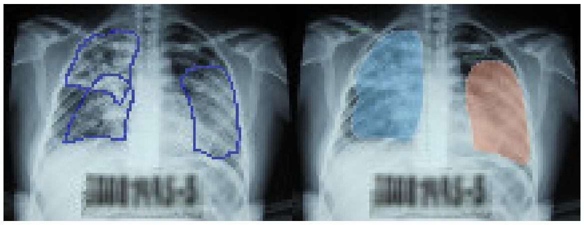


Fig. 6. Two ground truth areas match a single detection

The COCO metrics are reasonable approximations of performance, and are useful for automated analysis during training. However, to get a more precise measure of performance, the results of the Mask R-CNN network on the test set were evaluated manually, comparing the ground truth with the network results for each image, with the detection threshold set to 0.50. This allows us to correctly account for cases such as the two examples above, where the diseased regions have been correctly identified but would score low on the automated

[6] https://cocodataset.org/#detection-leaderboard

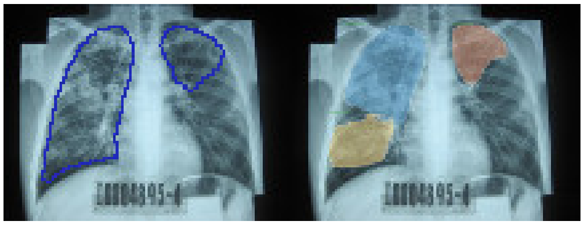


Fig. 7. One ground truth area matches two detections

COCO IoU metric. The results of this evaluation method are presented only for the Mask R-CNN network using the ResNeXt-101 backbone, as the other architectures follow the same trends.

The performance metrics for each of manifestation are shown in Table 6. Both precision and recall, and sensitivity and specificity are given, as the former is common in Computer Science literature while the latter is often used in medical and public health literature. Note that recall and sensitivity are two names for the same measure. Airspace Consolidation is recognized quite well, although with a high number of false positives. The other three manifestations have a higher proportion of false negatives, but the majority of errors are still false positives.

One item of note is that the precision given in Table 6 for Airspace Consolidation is lower than the AP50 values shown in Table 4. This is due to a quirk in the COCO evaluation method as implemented in the published COCO API[7]. The network may select multiple overlapping areas with different confidence levels over the same ground truth region. The COCO evaluation method counts each of these detections as correct if they exceed the IoU threshold. For Airspace Consolidation, which tends to have a large ground truth area, the network may produce multiple overlapping or concentric detections which exceed the IoU threshold. Counting all of these detections separately, rather than as a single positive case, artificially increases the COCO precision score. This can be seen in Fig. 8, which shows four detected areas covering two ground truth areas. This result should be counted as two correct detections rather than four. This problem does not appear to happen for the other manifestations, which tend to be much smaller, and is also unlikely to occur when dealing with natural images, where the object boundaries are more clearly defined.

Table 6. Precision, Recall / Sensitivity, and Specificity at a Confidence Threshold of 0.50 for Mask R-CNN

Manifestation	**Precision**	**Recall / Sensitivity**	**Specificity**
Airspace Consolidation	0.547	0.928	0.666
Cavitation	0.486	0.720	0.263
Lymphadenopathy	0.392	0.582	0.360
Pleural Effusion	0.580	0.607	0.610

[7] https://github.com/cocodataset/cocoapi

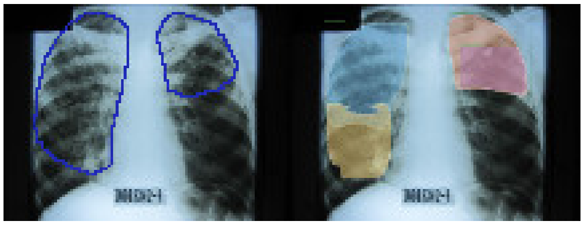


Fig. 8. Multiple areas detected for a single ground truth area

The Receiver Operating Characteristic (ROC) curves and AUC values for each manifestation are shown in Fig. 9. These results indicate that although there is room for improvement, the networks are learning to recognize and localize each manifestation at a rate far above what we would expect by random chance. The shape of the curve for Lymphadenopathy shows that there is a "sweet spot" where the performance is relatively good and other areas where it is poor. This and the large steps in the curve for Pleural Effusion is likely due to the limited number of instances.

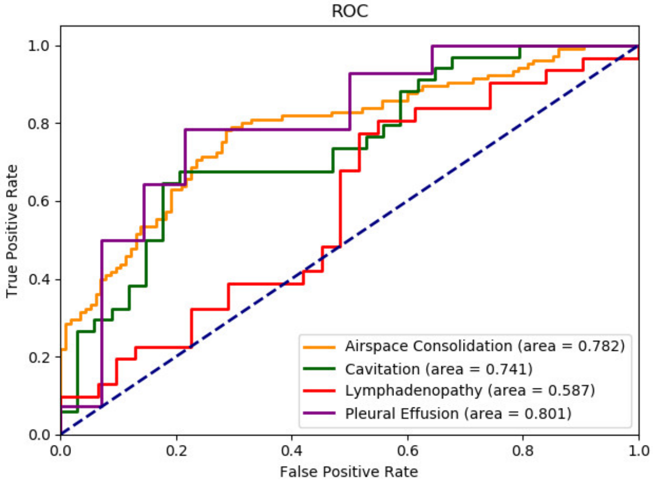


Fig. 9. ROC curves

In most cases, the network is able to closely match the ground truth area. The ground truth is defined by a user drawing a polygon around the affected area, and the edges usually provide a coarse outline. Often the network will have a tighter outline, as shown in Fig. 10.

The most common error we encounter is false positives, labeling normal areas as diseased. In some cases this occurs as detecting larger or additional areas in images, as in Fig. 11. In this image there is a small area of Airspace Consolidation

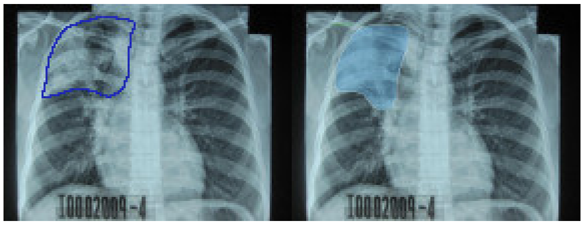


Fig. 10. Close match to ground truth

on the right lung. The network detected a larger region on the right lung and an additional small region on the left lung, both of which are false positives. Note that because this image contains a front-view CXR, the right lung is on the left side of the CXR and the left lung is on the right side.

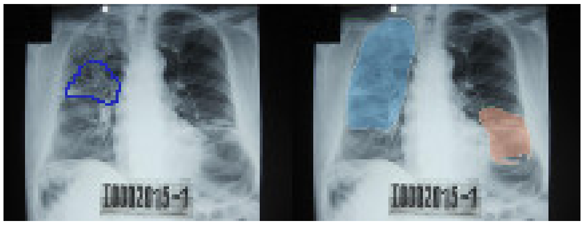


Fig. 11. Larger and additional false positive areas detected

This example highlights one challenge that arises from needing an experienced radiologist for the annotations. The machine learning researcher can compare the ground truth to the detected regions easily enough, but cannot provide a competent analysis of deviations. We need an expert to tell us if the additional areas are completely wrong, if they might be border line cases, or if the ground truth label was drawn strictly around the most affected area.

5.4 Additional Statistics

In addition to the results presented above, we collected seven additional statistics that shed some light on how useful the result of the network might be to a physician. These are:

- Whole image match - considers only the image level label and not the number or location of areas. This value is given for all images, positive samples (with the manifestation), and negative samples (without the manifestation). When correct, this at least lets the physician know whether or not there is a problem.
- Exact match - in this case the network result matches the ground truth exactly, i.e., there are no false positive or false negative areas. This value is given for all images and positive samples. (An exact match for a negative sample is the same as the whole image match.)

– Unmatched ground truth - images with some false negatives. This is the case we would like to avoid since we do not want to misdiagnose a sick patient.
– Unmatched detection - images with some false positives.

These values, shown in Table 7, are given as percentages of the number of images for each manifestation and are based on the manual evaluation. The extreme bias toward false positives over false negatives is clear. When a sample contains an area of interest the networks correctly find it, and usually do a good job of fitting the correct area. However, the networks also incorrectly mislabel normal areas at a high rate.

Table 7. Additional Statistics

	AC	**CA**	**LY**	**PE**
Whole image match all images	64.9%	58.2%	56.2%	66.9%
Whole image match positive samples	100.0%	97.8%	81.9%	69.2%
Whole image match negative samples	15.6%	28.3%	34.9%	67.1%
Exact match all images	47.8%	41.0%	36.1%	64.0%
Exact match positive samples	71.9%	56.3%	48.0%	62.3%
Unmatched ground truth	6.3%	17.1%	21.7%	18.3%
Unmatched detection	46.5%	49.3%	41.2%	16.1%

Looking closely at the results for Airspace Consolidation, the whole image match over all images shows the network was able to determine whether or not an image contains Airspace Consolidation with an accuracy of 64.9%. The breakdown between the positive and negative samples shows the strong bias to false positives, with the network identifying most images as containing Airspace Consolidation. The exact match for all images shows the percent of images where the network matched the number and location of each occurrence of Airspace Consolidation with the ground truth. On 47.8% of all the images, and 71.9% of the positive images, the network was able to perform the instance segmentation task perfectly, with no false positive or false negative areas. The unmatched ground truth value indicates that only 6.3% percent of the images annotated as having Airspace Consolidation contained areas which were not identified by the network. The unmatched detection value means that 46.5% of the images contain some false positive areas.

The results for the other three manifestations show similar trends, with less of a bias towards false positives. These manifestations tend to be much smaller than Airspace Cavitation and are therefore more difficult to detect, accounting for the higher number of false negatives. Of course in a deployment we can adjust the detection threshold to choose the balance between false positives and false negatives. For these tests the detection threshold was set at 0.5 in order to make useful comparisons between models.

Fig. 12 shows a typical sample of false negatives for Cavitation. Here the network detected two problem areas but missed two others identified by the expert. The small size of a typical cavitation instance and the fact that nearby instances

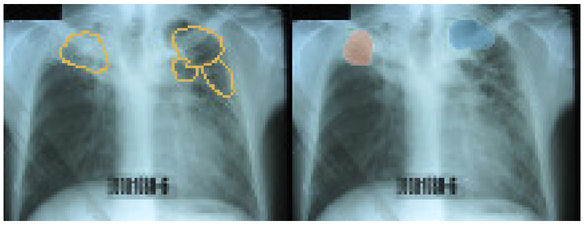


Fig. 12. Cavitation false negatives, two ground truth areas are not recognized

tend to be labeled separately (rather than being grouped as one instance) may contribute to the number of false negatives.

5.5 False Positive Reduction

The per manifestation heatmaps used during testing were able to effectively reduce the number of false positives for Lymphadenopathy and Pleural Effusion, the two manifestations which occur only in specific areas of the lungs. Of the 38 false positives identified by the model for Lymphadenopathy, 8 were filtered out using the heatmap. For Pleural Effusion, 4 out of 10 false positives were filtered out. This accounts for a significant number of false positives identified for these two manifestations.

Fig. 13 shows an example where an incorrect detection for Lymphadenopathy was filtered out based on the heatmap. The image shows the ground truth on the left, the unfiltered detection on the right, and the heatmap for Lymphadenopathy in the center. It is clear from the heatmap that this manifestation only occurs in the vertical center of the CXR, on either side of the spine. The detection near the collarbone is in an area where no ground truth instances are found in the training or validation sets and therefore could be removed from the set of detected instances. The other (correct) detection is kept as it overlaps an area in the heatmap with a high score.

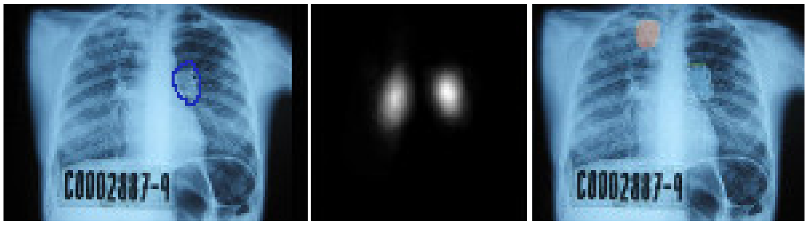


Fig. 13. Lymphadenopathy false positive reduction

The inclusion of the heatmap did not have any affect on the performance of either the Airspace Consolidation or Cavitation manifestations, which is expected as these can appear anywhere in the lung. There were also no additional false negatives for the Lymphadenopathy and Pleural Effusion manifestations. Thus, there were no negative affects to adding this processing.

6 Discussion

Our results show that using a well annotated dataset we can construct models for instance segmentation of TB manifestations in CXRs with performance measures that are reasonable given the size of our dataset and the difficulty of the task. However, the question of how such models can be used in a real world setting remains, especially given the current level of performance. The precision and recall (sensitivity and specificity) values both would need to be higher for our models to be used as a stand alone diagnostic aid. There are some approaches we can take to improve this situation.

The first improvement we can make is to increase the performance of our models by increasing the size of our dataset. More data typically translates into better performance for deep learning models and the size of our dataset is very small compared to datasets use for instance segmentation on natural images such as COCO [19] or LVIS [9] which each have over 100,000 images. Increasing the number of annotated CXRs should increase both our precision and recall and the number of manifestations we can reliably recognize.

However, even if we are able to achieve performance rivaling the state of the art on natural image datasets, the accuracy would still be too low. At the time of this writing the best results posted on the COCO leaderboard show an AP50 score of 0.766. This score indicates that the model missed almost a quarter of the instances, a result that would not be sufficient in a diagnostic setting. While the state of the art performance on the instance segmentation task alone does not reach the level we would like, we may be able to use the instance segmentation models in conjunction with whole image models to construct a reasonable screening system. [22] reports an AUC of 0.938 on whole image classification of the TBX11K dataset, distinguishing between patients with and without TB. The CheXpert challenge[8], which aims to identify five specific pathologies in CXRs using the CheXpert dataset, shows a leaderboad with many models having an AUC over 0.90. These examples indicate that whole image classification may be a good first step in identifying TB, either generally or from specific abnormalities. We may be able to use instance segmentation or object detection as a second step to provide a diagnosing physician with accurate and detailed analysis.

Finally, we may be able to make use of domain knowledge to improve our performance. Using heatmaps for false positive reduction is one example of this. There may be additional knowledge about particular manifestations or associations between manifestations that we can take advantage of to increase our performance.

7 Conclusions and Future Work

Tuberculosis, a preventable and curable disease, remains one of the leading causes of death world wide. One approach to dealing with this issue is decreasing the time required to perform a diagnosis by providing physicians with proper tools and accurate information. This work investigated the possibility of using CNNs

[8] https://stanfordmlgroup.github.io/competitions/chexpert/

to locate manifestations of TB in CXRs. We have shown that with high quality, object level annotations of CXRs, it is possible to train models to perform this task. This has the potential of providing diagnosing physicians with tools to help increase their efficiency, especially in high TB regions in LMICs.

We compared the performance of Faster R-CNN, Mask R-CNN, Cascade versions of each, and SOLOv2, on a new dataset with object level annotations of four manifestations associated with TB. Each network was able to learn to correctly detect or segment the areas of interest, with varying degrees of success based on the manifestation. Our results show that Faster R-CNN and Mask R-CNN had similar performance at AP50, with Faster R-CNN performing better at the overall mAP. The addition of the Cascade network did not improve our results, which is possibly due to our small dataset. The two-stage networks outperformed the single-stage SOLOv2 network for this task.

Some manifestations are strongly associated with a particular area of the lung. We showed that this domain-specific knowledge can be used in the form of a heatmap to successfully reduce the number of false positives.

To achieve a higher level of performance we need to continue the annotation work, increasing both the number of manifestations recognized and the number of labeled instances of each. As with most deep learning approaches, more data should yield better performance. The results presented here support the case for spending the resources in creating a larger dataset with object level annotations of TB manifestations. Once a well annotated dataset of sufficient size is available the next step is to address the question of how to best adapt a CAD system for screening and triage that could be efficiently integrated into existing TB health systems in LMICs. This is quite relevant in today's Covid-19 era as not only TB control programs worldwide have been severely affected by the pandemic but also for the potential to also contribute to Covid-19 control in both the Global North and the South.

References

1. Adfas, D.: World health organization tuberculosis fact sheet (2019), `https://www.who.int/en/news-room/fact-sheets/detail/tuberculosis`
2. Alcantara, M.F., Cao, Y., Liu, C., Liu, B., et al.: Improving tuberculosis diagnostics using deep learning and mobile health technologies among resource-poor communities in Perú. Smart Health **1-2**, 66–76 (2017)
3. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (2018)
4. Cao, Y., et al.: Improving tuberculosis diagnostics using deep learning and mobile health technologies among resource-poor and marginalized communities. In: 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE). pp. 274–281 (June 2016). https://doi.org/10.1109/CHASE.2016.18
5. Chen, K., et al.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Computer Vision and Pattern Recognition. pp. 248–255 (2009)

7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014)
8. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
9. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5351–5359 (06 2019). https://doi.org/10.1109/CVPR.2019.00550
10. Hardy, M., Harvey, H.: Artificial intelligence in diagnostic imaging: impact on the radiography profession. The British Journal of Radiology **93** (2020). https://doi.org/10.1259/bjr.20190840, `https://doi.org/10.1259/bjr.20190840`, PMID: 31821024
11. Harris, M., et al.: A systematic review of the diagnostic accuracy of artificial intelligence-based computer programs to analyze chest x-rays for pulmonary tuberculosis. PLoS ONE **14(9)** (2019), `https://doi.org/10.1371/journal.pone.0221339`
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. CoRR **abs/1703.06870** (2017), `http://arxiv.org/abs/1703.06870`
14. Hwang, S., Kim, H.E., Jeong, J., Kim, H.J.: A novel approach for tuberculosis screening based on deep convolutional neural networks. In: Tourassi, G.D., III, S.G.A. (eds.) Medical Imaging 2016: Computer-Aided Diagnosis. vol. 9785, pp. 750–757. International Society for Optics and Photonics, SPIE (2016). https://doi.org/10.1117/12.2216198, `https://doi.org/10.1117/12.2216198`
15. Islam, M.T., Aowal, M.A., Minhaz, A.T., Ashraf, K.: Abnormality detection and localization in chest x-rays using deep convolutional neural networks. CoRR **abs/1705.09850** (2017), `http://arxiv.org/abs/1705.09850`
16. Jaeger, S., Candemir, S., Antani, S., Wáng, Y., Lu, P., Thoma, G.: Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. Quantitative imaging in medicine and surgery **4(6)**, 475–477 (2014)
17. Jaeger, S., et al.: Automatic tuberculosis screening using chest radiographs. IEEE transactions on medical imaging **33(2)**, 233–245 (2014)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems pp. 1097–1105 (2012)
19. Lin, T., et al.: Microsoft COCO: common objects in context. CoRR **abs/1405.0312** (2014), `http://arxiv.org/abs/1405.0312`
20. Liu, C., et al.: Tx-cnn: Detecting tuberculosis in chest x-ray images using convolutional neural network. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 2314–2318 (Sep 2017). https://doi.org/10.1109/ICIP.2017.8296695
21. Liu, J., Lian, J., Yu, Y.: ChestX-Det10: Chest x-ray dataset on detection of thoracic abnormalities (2020), `https://arxiv.org/abs/2006.10550`
22. Liu, Y., Wu, Y.H., Ban, Y., Wang, H., Cheng, M.M.: Rethinking computer-aided tuberculosis diagnosis. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2643–2652 (2020)
23. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3431–3440 (2015)

24. Meraj, S.S., et al.: Detection of pulmonary tuberculosis manifestation in chest x-rays using different convolutional neural network (cnn) models. International Journal of Engineering and Advanced Technology **9** (12 2019)
25. Paszke, A., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
26. Qin, C., Yao, D., Shi, Y., Song, Z.: Computer-aided detection in chest radiography based on artificial intelligence: a survey. BioMedical Engineering OnLine **17** (2018). https://doi.org/10.1186/s12938-018-0544-y, https://doi.org/10.1186/s12938-018-0544-y
27. Rajpurkar, P., et al.: Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. CoRR **abs/1711.05225** (2017), http://arxiv.org/abs/1711.05225
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell (2017)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
30. Stirenko, S., et al.: Chest x-ray analysis of tuberculosis by deep learning with segmentation and augmentation. CoRR **abs/1803.01199** (2018), http://arxiv.org/abs/1803.01199
31. Szegedy, C., Liu, W., Jia, Y., Sermanet, et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9 (2015)
32. Ugarte-Gil, C., et al.: Implementing a socio-technical system for computer-aided tuberculosis diagnosis in Peru: A field trial among health professionals in resource-constraint settings. Health Informatics Journal (2020), https://doi.org/10.1177/1460458220938535
33. Wang, X., Peng, Y., Lu, L., Lu, Z., M.Bagheri, Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3462–3471 (2017)
34. Wang, X., Kong, T., Shen, C., Jiang, Y., Li, L.: Solo: Segmenting objects by locations (2020), https://arxiv.org/abs/1912.04488
35. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: Solov2: Dynamic, faster and stronger (2020), https://arxiv.org/abs/2003.10152
36. World health organization global tuberculosis report (2019), https://www.who.int/tb/publications/global_report/en/
37. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)
38. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5987–5995 (07 2017). https://doi.org/10.1109/CVPR.2017.634