Comparison of Object Detection and Instance Segmentation Methods in Chest X-rays

Terence Griffin
Department of Computer Science
University of Massachusetts
Lowell, MA, USA
terence_griffin@student.uml.edu

***Abstract*—This project will investigate object detection and instance segmentation methods for identifying abnormalities in chest X-rays (CXR). The aim is to determine if there is a method which will outperform Mask R-CNN and Faster R-CNN for this specific task. We compare the performance on the object detection task of Faster R-CNN, SSD, RetinaNet, YOYOv3, and FCOS, and the performance on the instance segmentation task of Mask R-CNN, BlendMask, and SOLOv2. Four chest X-ray datasets are used for this evaluation: UML/Perú, ChestX-ray14, ChestX-Det10, TBX11K.**

***Index Terms*—Object Detection, Instance Segmentation, Medical Imaging, X-ray, Deep Learning**

I. Introduction

This project compares several deep learning networks for object detection and instance segmentation of lung diseases and abnormalities in chest X-ray (CXR) images. We are particularly interested in lung abnormalities indicative of Tuberculosis (TB), however given the sparsity of labeled data, we also include results using datasets for general pulmonary issues.

TB is an infectious disease affecting millions of patients each year, especially in Low and Middle Income Countries (LMIC) with limited heath care resources. 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 [2].

One approach to help limit the impact and spread of this disease is constructing tools for the automated screening of TB which assist overburdened physicians and nurses to more quickly arrive at a diagnosis and start treatment if necessary. The work presented here is part of an ongoing project to improve the speed and accuracy of TB screening, described in [3]–[5]. The work presented here is focused on evaluating several state of the art approaches to object detection and instance segmentation in order to determine which approaches perform best on the available CXR datasets. These results can then be used to guide a selection of models to incorporate into an automated screening system which provide a reasonable balance of accuracy and detailed information.

Most published research on these two tasks provides evaluation of models on datasets composed of natural images, such as COCO [6], ImageNet [7], or LVIS [8]. The identification and localization of pulmonary diseases and abnormalities in CXRs is different from that of objects in natural images in a number of ways:

- Natural images have a distinct hard boundary, whereas lung abnormalities may not.
- All CXRs have a lot of commonality, having the same overall appearance, with the significant differences being in the details. Natural images can show a wide variety of difference scenes.
- CXRs are grayscale images (even if encoded as RGB). The color variations in natural images may be used for the detection and segmentation tasks in ways not applicable to CXRs.
- Identifying objects in natural images is an easy task for humans. Finding abnormalities in CXRs is a more difficult task, requiring special training.

Given these differences, it is reasonable to assume that the best performing methods on natural images may not be the best methods for CXRs. Evaluating strong models on available CXR datasets containing object level annotations may provide some insight into how the differences between natural images and CXRs impact model performance.

II. Related Work

A. Object Detection

The object detection tasks involves taking an image as input and determining the class and location of objects in the image, using bounding boxes to describe the object location. Since the emergence of convolutional neural networks (CNNs) as the preferred tool for image classification in the early 2010's [9] many methods for using CNNs for the related task of object detection have been proposed.

One of the early strong performers is Faster R-CNN [10], [11], first published in 2016. Like other models for object detection and instance segmentation, Faster R-CNN uses the convolutional layers of a network designed for image classification as a backbone network to extract feature maps from the input image. Theses feature maps are then used for localizing and classifying objects in the image.

Faster R-CNN has a two-stage structure (see Fig. 1). The first stage is a Region Proposal Network (RPN), which learns how to identify candidate Regions for Interest (ROIs) containing objects. The second stage, the so-called "head network",

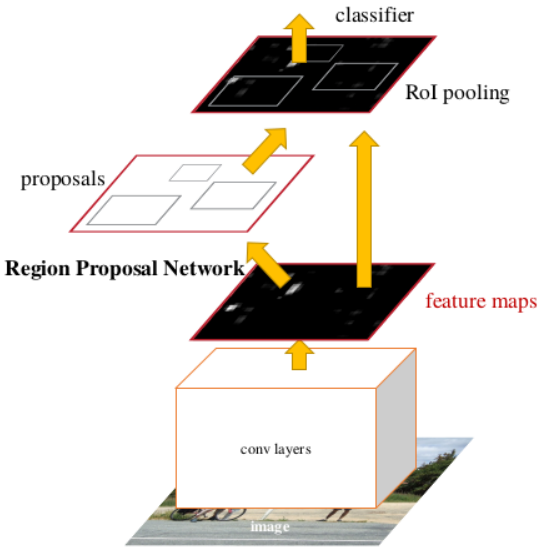

Fig. 1. Faster R-CNN

uses the proposed ROIs as input and uses parallel networks to identify the class and bounding box for objects in the input image. The RPN and head network are able to share the feature maps from the backbone network, producing a model that is both more accurate and more efficient than previous approaches.

Although an improvement in speed compared with contemporary approaches, Faster R-CNN is not sufficiently fast for real-time applications. Several single-stage networks have been proposed which use significantly fewer parameters than Faster R-CNN, allowing for inference times within the real-time range (near 30 fps). The reported performance of single-stage networks has increased to within or slightly beyond that of Faster R-CNN due to specific optimizations and careful tuning.

Real-time performance is not important for our task. However, it is possible that the simpler structure of single-stage networks, or the affect of the optimizations used, provide a benefit for the analysis of CXRs. Since the task of object detection on CXRs is in some ways simpler and in some ways harder than that for natural images it is unclear how single-stage networks will perform compared with a two-stage network.

YOLO (You Only Look Once) [12]–[15] is a series of progressively more powerful object detectors based on a common foundation. YOLO divides the image into a fixed size grid and then predicts a single class and a small number of bounding boxes (2 in the original paper) for an object whose center lines in each grid cell (see Fig. 2). Using a grid system, rather than an RPN or other proposal mechanism which produces are large number of candidates, results in a very efficient network in terms of inference time. Subsequent iterations of the YOLO architecture provide improvements including a new backbone network, Darknet, using a Feature Pyramid Network (FPN), Mish activation, and data augmentation techniques.

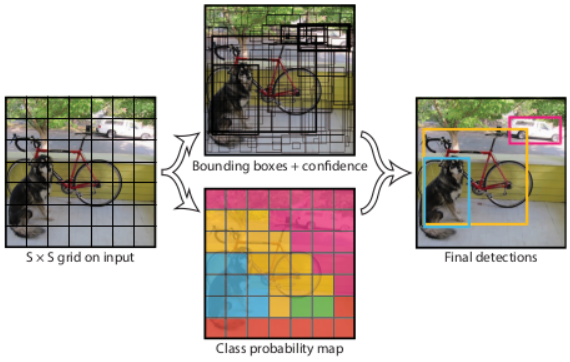

Fig. 2. YOLO

Single Shot Detector (SSD) [5] is a relatively simple approach for object detection using a single network. Instead of using a RPN to propose regions, SSD uses a small fixed set of candidate scales and aspect ratios, which operate over discritized feature maps. Fig. 3 shows the network structure based on a VGG backbone, the added convolutional layers provide Multiple feature maps are used, at different scales, proving some scale invariance.. The network predicts a score for the object class and final bounding box adjustment for each proposed box. The small number of proposed boxes allows the network to be fast during inference. The accuracy of the network depends significantly on how well the fixed scales and aspect ratios for the proposed boxes matches the relative sizes of the objects of interest in the images. Through careful tuning, SSD can be made to perform well for a chosen task.

RetinaNet [16] seeks to improve the performance of single-stage detectors through the use of a new loss function. One advantage that two-stage detectors have is that the RPN learns how to identify good bounding box candidates, which allows the head networks to see a reasonable balance between ROIs with and without ground truth objects. The head networks in single-stage models, on the other hand, see a much larger proportion of negative samples. The Focal Loss function, introduced in [16], uses a tunable focusing parameter γ to put more weight on difficult examples. The focal loss, with an α balancing parameter is:

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

Combining focal loss with a standard backbone such as ResNet, and a feature pyramid network to provide scale invariance (see Fig. 4) allows a single-stage network to achieve performance on par with Faster R-CNN [16].

Faster R-CNN, YOLO[1], SSD, and RetinaNet all depend on pre-defined anchor boxes for identifying objects within an image. FCOS [17] (Fully Convolutional One-Stage Object Detection) uses a different approach, predicting class, centerness value, and boudning box values for each pixel in each

[1]YOLOv1 did not use anchor boxes, this was changed in YOLOv2 to improve performance

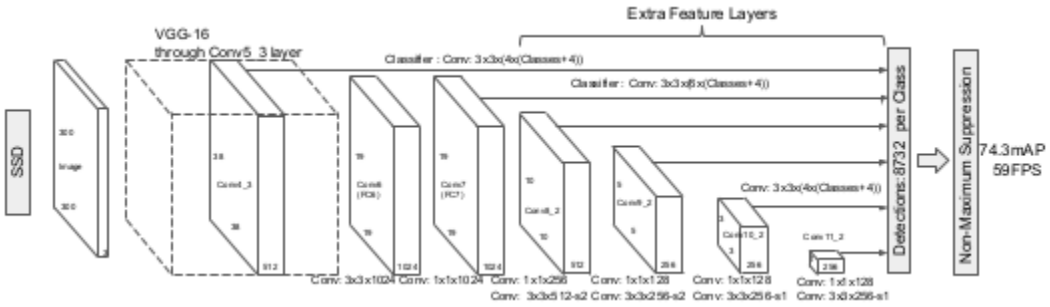

Fig. 3. SSD

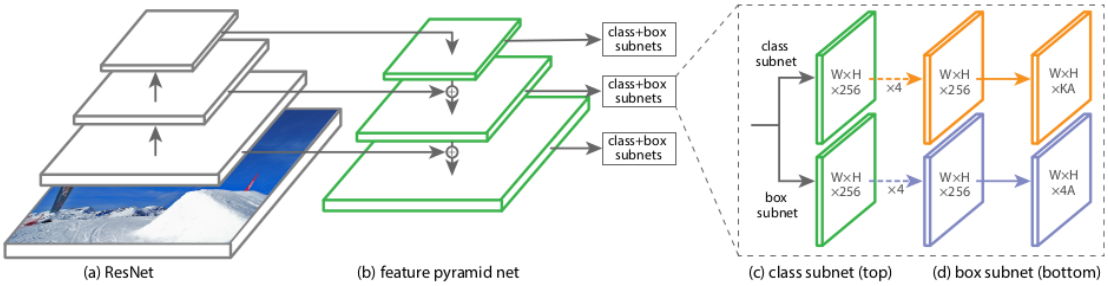

Fig. 4. RetinaNet

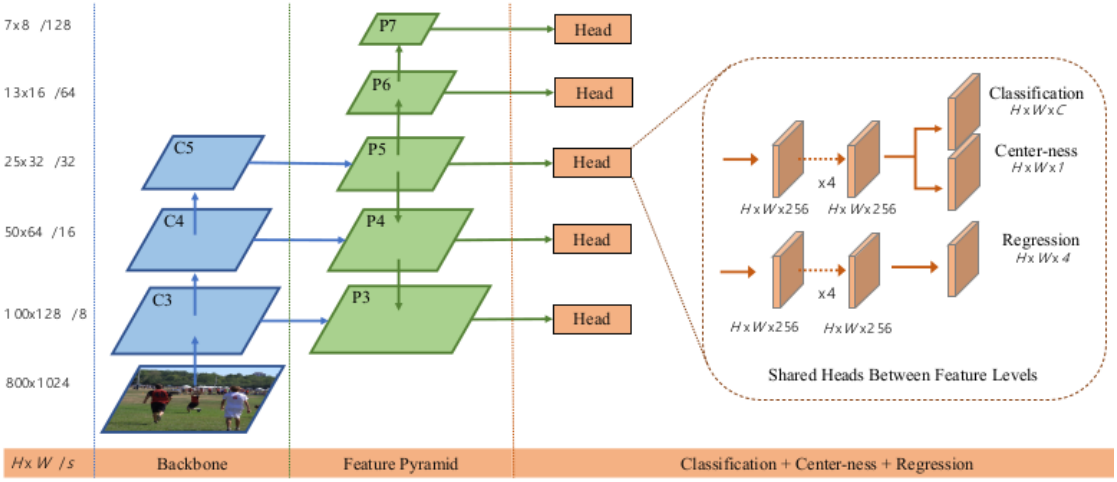

Fig. 5. FCOS

feature map. The structure is shown in Fig. 5. The center-ness branch predicts the deviation from a pixel to the center of its corresponding bounding box, and it is this addition which allows the network to perform as well as anchor-based approaches. The results from the three branches need to be processed by a Non-Maximum Suppresssion (NMS) step to produce the final list of predictions.

B. Instance Segmentation

The instance segmentation task predicts a pixel-wise mask for each object, rather than a bounding box. The first approach we look at is Mask R-CNN [18], which builds on the Faster

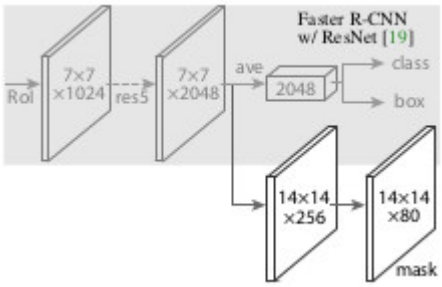

Fig. 6. Mask R-CNN head network

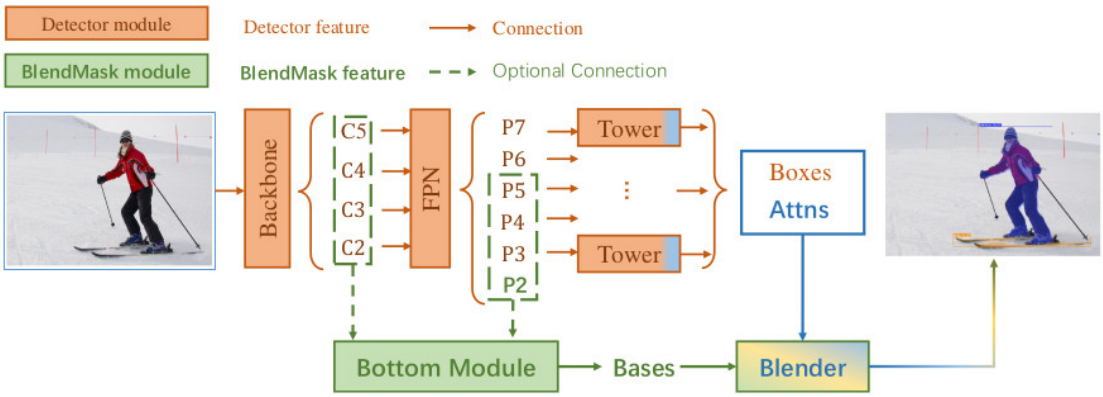

Fig. 7. BlendMask

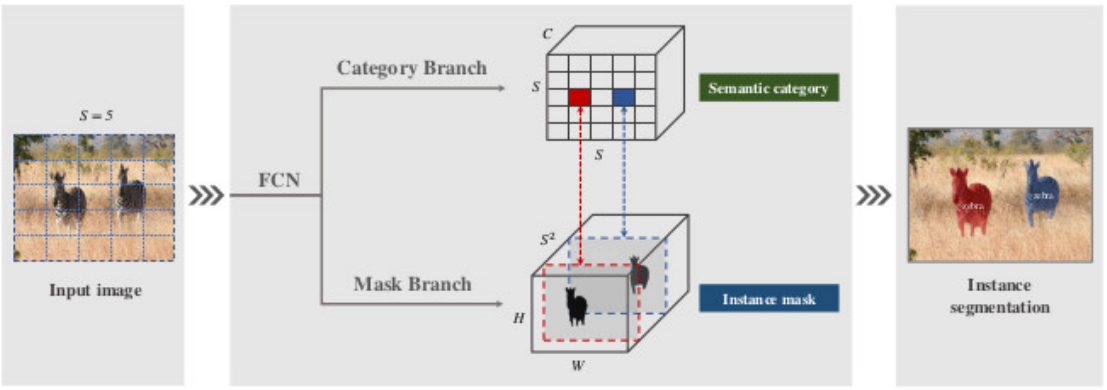

Fig. 8. SOLO

R-CNN architecture. Mask R-CNN adds a third branch to the head network, predicting a mask for an object within the ROI, as shown in Fig. 6. The mask branch produces a binary mask for each class (80 for the COCO dataset). This mask is then upsampled from the dimensions of the feature map used to create the ROI to the matching dimensions in the original image.

BlendMask [19] builds on the FCOS object detection model. Using a similar anchor-free approach, an additional head branch (label "Tower" in the figure) is used to predict the mask. A novel blend module combines the results from the head networks with features from the backbone network to produce accurate pixel-wise masks (see Fig. 7). [19] reports accuracy results at the same level as Mask R-CNN, while also showing a significant improvement in speed.

SOLO [20], [21] (Segmenting Objects by Locations) takes a grid approach, similar to YOLO. The basic structure is shown in Fig. 8. The input image is divided into an $S \times S$ grid, and each grid cell location is responsible for predicting the class and mask for a single object.

C. Object Detection in CXRs

As stated earlier, most work on object detection has been done using datasets of natural images. There are three sources presenting results using CXR datasets. The ChestX-ray14 [22] dataset contains 112,120 images from 32,717 patients with whole image annotations for 14 pulmonary diseases. Object level (bounding box) annotations for eight classes are given for 880 images. The eight classes are: Atelectasis, Cardiomegaly, Effusion, Infiltrate, Mass, Nodule, Pneumonia, and Pneumothorax, of which Effusion and Infiltrate are related to TB and so are of particular interest to us. The small number of images and number of instances of each class (less than 200) limit how well a network can learn using this dataset.

The ChestX-Det10 [23] dataset was created from 3,543 images from the ChestX-ray14 dataset. Object level annotations are provided for ten diseases: Atelectasis, Calcification, Consolidation, Effusion, Emphysema, Fibrosis, Fracture, Mass, Nodule, and Pneumothorax. Note that Consolidation and Infiltrate are two terms for similar conditions, and of these classes Consolidation and Effusion are related to TB. The larger size of this dataset allows us to achieve better performance than with the original ChestX-ray14 dataset annotations.

The TBX11K [24] dataset consists of 11,200 images with whole image labels for four classes: Healthy, Active TB, Latent TB, and Sick Non-TB. There are bounding box annotations for the Active TB and Latent TB instances, with 924 and 212 images respectively. While this dataset lacks

TABLE I
CLASS-SPECIFIC OBJECT DETECTION RESULTS

	UML/Perú			ChestX-ray14			ChestX-Det10			TBX11K		
	mAP	AP50	AR	mAP	AP50	AR	mAP	AP50	AR	mAP	AP50	AR
Faster R-CNN	0.207	0.534	0.451	0.180	0.457	0.393	0.201	0.515	0.447	0.190	0.501	0.411
SSD512	0.186	0.386	0.459	0.147	0.327	0.396	0.179	0.377	0.436	0.161	0.400	0.401
RetinaNet	0.216	0.559	0.477	0.198	0.487	0.412	0.223	0.537	0.470	0.192	0.436	0.478
YOLOv3-608	0.196	0.395	0.390	0.168	0.371	0.363	0.188	0.400	0.396	0.181	0.411	0.423
FCOS	0.189	0.383	0.455	0.143	0.325	0.381	0.182	0.379	0.435	0.158	0. 401	0.410

TABLE II
CLASS-AGNOSTIC OBJECT DETECTION RESULTS

	UML/Perú			ChestX-ray14			ChestX-Det10			TBX11K		
	mAP	AP50	AR	mAP	AP50	AR	mAP	AP50	AR	mAP	AP50	AR
Faster R-CNN	0.247	0.575	0.512	0.190	0.462	0.421	0.222	0.534	0.468	0.210	0.521	0.431
SSD512	0.193	0.393	0.471	0.158	0.331	0.402	0.186	0.395	0.456	0.173	0.421	0.409
RetinaNet	0.265	0.595	0.525	0.217	0.490	0.433	0.241	0.553	0.483	0.205	0.448	0.495
YOLOv3-608	0.201	0.452	0.493	0.170	0.358	0.402	0.193	0.400	0.462	0.193	0.437	0.425
FCOS	0.195	0.396	0.475	0.156	0.328	0.399	0.190	0.397	0.458	0.177	0.420	0.412

TABLE III
INSTANCE SEGMENTATION RESULTS

	UML/Perú		
	mAP	AP50	AR
Mask R-CNN	0.197	0.534	0.461
BlendMask	0.112	0.287	0.389
SOLOv2	0.135	0.325	0.383

the detail of the particular pathology it does provide for a relatively large number of instances related to TB. In addition to the usual class specific results, this paper includes results for class-agnostic detection.

Our own UML/Perú dataset contains mask annotations for 1,186 images, using 11 pathologies. Of these images there are four classes with sufficient number of instances to train an instance segmentation model: Airspace Consolidation, Cavitation, Lymphadenopathy, and Pleural Effusion. Of the four datasets avialable this is the only one which provides mask annotations, and thus can be used for instance segmentation. We are able to create bounding boxes based on the mask annotations, so we can use this dataset for the object detection tests as well.

III. APPROACH

For the object detection task the following network architectures were tested: Faster R-CNN [10], SSD [25], RetinaNet [16], and YOLOv3 [14]. Each network was tested with the four available datasets for both the class-specific and class-agnostic detection tasks. Each dataset was split into train, validation, and test sets, using an 80%/10%/10% split.

For the instance segmentation task only the UML/Peru dataset has the necessary annotations. The Mask R-CNN [18], BlendMask [19], and SOLOv2 [21] architectures were tested. A class-agnostic test was not done using these networks, as it is less clear whether this makes sense for a mask.

The Faster R-CNN, Mask R-CNN, RetinaNet, and SOLOv2 networks were tested with ResNet [26] and ResNeXt [27] backbone networks of size 50, 101, and 152. BlendMask was tested with ResNet 50 and 101 backbones, as ResNeXt was not supported in the AdelaiDet library and time didn't allow for adding this network. Since we are less concerned with the networks, we only tested the larger SSD512 and YOLOv3-608 networks. Roughly a week of calendar time was devoted to tuning the hyperparameters for each network.

The implementation for the networks is based on MMDetection [28], with the exception of BlendMask which uses the implementation reference in [19]. 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 two and seven hours, depending on the network architecture and the hyperparameters.

IV. RESULTS

For the networks using a ResNet or ResNeXt backbone, the ResNeXt-101 network had the best performance in all cases. The following results are all for networks using this backbone network. BlendMask worked best with the ResNet-101 backbone. Results are presented using the COCO [6] metrics for mean average precision (mAP), average precision at an intersection over union threshold of 0.50 (AP50), and average recall using 300 detections (AR).

Table I shows the results for the class-specific object detection task. RetinaNet performs slightly better than Faster R-CNN across all four datasets, with YOLOv3 and SDD512 performing worse. FCOS, the only anchor-less network, performed similar to SSD512, and well below the leaders.

The performance of the four networks is consistent across each datasets. All networks performed best on the UML/Peru

dataset, had close to the same performance on ChestX-Det10 and TBX11K, and performed worst on ChestX-ray14.

Table II shows the results for the class-agnostic task. The results in all cases are slightly better than for the class-specific task. The largest gain in AP50 score was 0.035 on the UML/Perú data. The trends between the different models and datasets are the same as for the class-specific task.

Table III shows the results for the instance segmentation task. Here we are limited to the UML/Peru dataset as it is the only one with mask annotations. Mask R-CNN outperforms the other two networks, with SOLOv2 having slightly better performance than BlendMask.

V. Discussion

This investigation started with some knowledge of the performance of the Faster R-CNN and Mask R-CNN networks on the UML/Perú dataset. The goal of this work was to see if one of the more recent architectures would outperform these networks. For the object detection task we found that RetinaNet was the only network tested which surpassed Faster R-CNN. Both SSD and FCOS are significantly simpler than Faster R-CNN and had relatively poor performance, leading to the conclusion that our task requires a more complex network.

The networks tested showed consistent trends across the four datasets. The relative performance does not track with the size of the dataset. ChestX-ray14 is the smallest dataset and does show the worst performance. However, the UML/Perú data is smaller than both ChestX-Det10 and TBX11K but has better results. One possible explanation is that that annotations were made by a single TB expert and may be more self-consistent than the other datasets.

The class-agnostic testing showed better results in all cases but only be a small amount. This indicates that the task of distinguishing between classes is less difficult than the localization task. There was not a significant difference in the gain from the ChestX-Det10 dataset, which has ten classes, and the TBX11K dataset, which only has two classes. If class labeling was the difficult part of the task then we would expect to see more improvement in the datasets with a larger number of classes.

For the instance segmentation task Mask R-CNN outperforms the other approaches significantly. As BlendMask is based on FCOS, and FCOS did not perform well on the object detection task, it is reasonable that BlendMask also would struggle. There was a large difference in performance between SOLOv2 and Mask R-CNN, much more than would be suggested by [21]. This may be due to the difficulty of the task, or the size of the objects we are detecting. SOLOv2's performance is best with larger objects [21], and struggles more with small objects.

The performance of Mask R-CNN on the instance segmentation task was the same as Faster R-CNN for AP50 and slightly lower for mAP. RetinaNet was able to achieve better scores for both AP50 and mAP.

Of the backbone networks tested, the more recent, and larger, ResNeXt backbone performed best, suggesting that even more powerful image classification backbone networks could lead to improved performance.

The choice of architectures and backbone networks was limited by time. Future work should include testing the YOLOv4 [15] network, and in particular performing a similar ablation study to evaluate the impact of the various special features tested. The Efficient-Det [29] network is also worth investigating.

VI. Conclusion

For this project w investigated the performance of different architecture for object detection and instance segmentation on four different CXR datasets. Our results show that the two-stage detectors for the most part outperform the most recent architectures, which tend to be simpler and are designed with more emphasis on speed. In the object detection task the best performer was a single-stage network, RetinaNet, which shows that there may be improvements that could be made over the two-stage approach.

Mask R-CNN showed the best performance for the instance segmentation segmentation task among the three architectures tested. The performance of the best object detection network was higher than for instance segmentation, indicating that finding bounding boxes may be significantly easier than constructing masks.

The performance across the four datasets tested varied but were consistent between models. There were no cases where a particular model worked relatively better or worse with a particular dataset.

References

[1] D. adfas, "World health organization tuberculosis fact sheet," 2019. [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/tuberculosis

[2] "World health organization global tuberculosis report," 2019. [Online]. Available: https://www.who.int/tb/publications/global_report/en/

[3] Y. Cao *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)*, June 2016, pp. 274–281.

[4] C. Liu *et al.*, "Tx-cnn: Detecting tuberculosis in chest x-ray images using convolutional neural network," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 2314–2318.

[5] C. Ugarte-Gil *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. [Online]. Available: https://doi.org/10.1177/1460458220938535

[6] T. Lin *et al.*, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[8] A. Gupta, P. Dollar, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," in *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2019, pp. 5351–5359.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell*, 2017.

[11] R. Girshick, "Fast r-cnn," *ICCV*, 2015.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[13] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[14] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[15] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 07 2018.

[17] Z. Tian, C. Shen, H. Chen, and H. Tong, "Fcos: Fully convolutional one-stage object detection," in *IEEE International Conference on Computer Vision*, 10 2019, pp. 9626–9635.

[18] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 06 2018.

[19] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," 2020. [Online]. Available: https://arxiv.org/abs/2001.00309

[20] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "Solo: Segmenting objects by locations," 2020. [Online]. Available: https://arxiv.org/abs/1912.04488

[21] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "Solov2: Dynamic, faster and stronger," 2020. [Online]. Available: https://arxiv.org/abs/2003.10152

[22] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3462–3471.

[23] J. Liu, J. Lian, and Y. Yu, "ChestX-Det10: Chest x-ray dataset on detection of thoracic abnormalities," 2020. [Online]. Available: https://arxiv.org/abs/2006.10550

[24] Y. Liu, Y. H. Wu, Y. Ban, H. Wang, and M. M. Cheng, "Rethinking computer-aided tuberculosis diagnosis," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2643–2652.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*, vol. 9905, 10 2016, pp. 21–37.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[27] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 07 2017, pp. 5987–5995.

[28] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[29] M. Tan, R. Pang, and Q. Le, "EfficientDet: Scalable and efficient object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 06 2020, pp. 10 778–10 787.