Pneumonia Detection in Chest X-Rays using RCNN and YOLOv5

Saswato Bhattacharyya, Apurva Nehru and Gaurav Thorat

College Of Engineering, Northeastern University

May 6, 2021

Abstract

In 2015 alone, 920,000 children under the age of 5 died from pneumonia. In the US, pneumonia is responsible for over 500,000 visits to ER and over 50,000 deaths in 2015. The diagnosis of pneumonia on CXR is complicated because of several other factors such as fluid overload (pulmonary edema in the lungs) or bleeding. In 2015 alone, 920,000 children under the age of 5 died from pneumonia. In the US, pneumonia is responsible for over 500,000 visits to ER and over 50,000 deaths in 2015. Pneumonia is diagnosed using chest x-ray (CXR). Diagnosis of pneumonia on CXR is complicated because of several other underlying conditions such as fluid overload or bleeding etc.

In this project we are trying to predict whether a given CXR can be predicted as inflicted with pneumonia or not using YOLOv5 and Mask-RCNN.

Introduction

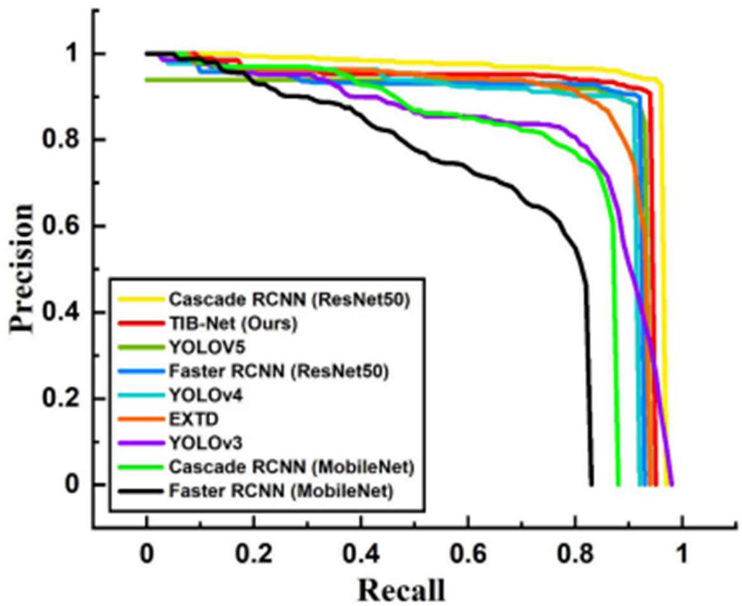
Though common, accurately diagnosing pneumonia is difficult. It calls for review of a chest x-ray (CXR) by trained specialists and confirmation through review of clinical history, vital signs, and laboratory exams. CXR's is also the most performed diagnostic imaging study. Pneumonia usually exhibits as an area or areas of increased opacity on CXR. However, the diagnosis of pneumonia on CXR is complicated because of several other factors such as, fluid overload (pulmonary edema) in the lungs, bleeding, volume loss (atelectasis or collapse), lung cancer, or post-radiation or surgical changes. When available, comparison of CXRs of the patient taken at different time points and correlation with clinical symptoms and history are helpful in making the diagnosis.

This project is based on pneumonia detection using RCNN and YOLOv5. We compared two models, Mask - RCNN and YOLOv5. Both the models we use are pretrained. Our dataset consisted of labelled images of CXR's in DCM format. We convert the same into jpg format for YOLOv5. We then trained YOLOv5 on PyTorch and Mask - RCNN on TensorFlow's Object Detection API. We then tested both the models and compared their results. We find that the mAP of Mask - RCNN was 98.15 and YOLOv5 was 57.5.

3.Proposed Method

To detect presence of pneumonia in CXR, using CNN we can accomplish this by training a network on labelled images that accurately label the CXR in all the images. We first gathered a dataset. Then we chose two models for this task:

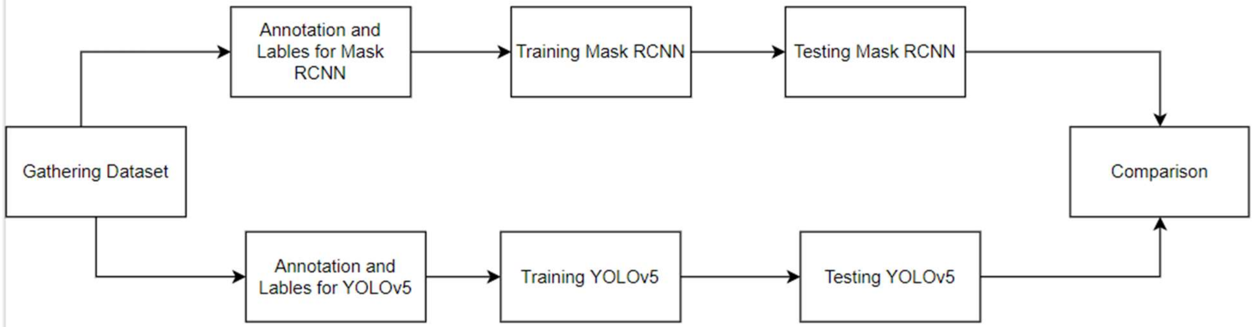
First being Mask - RCNN and the second, YOLOv5.



We chose the Mask - RCNN which is built on top of Faster RCNN and YOLO v5 (built on top of YOLOv4) because its precision and recall was one of the highest as shown in the above figure.

We further convert the images to JPG format from DCM and then train the images with respective models (YOLOv5 and Mask-RCNN). Then we test and evaluate which is better.

Project Workflow

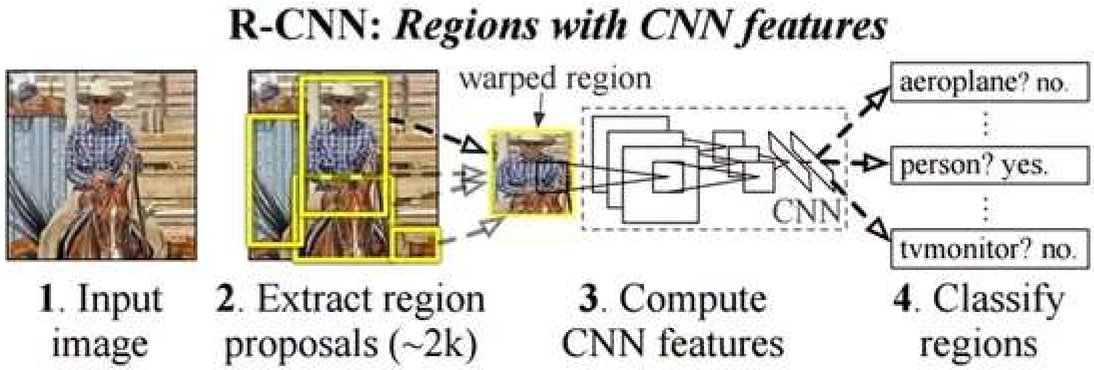


3.1 Introduction to Mask – RCNN

Region Based Convolution Neural Network - RCNN

The purpose of R-CNNs(Region Based Convolution Neural Network) is to solve the problem of object detection. Given a certain image, we want to be able to draw bounding boxes over all of the objects.

Selective search is used in particular for RCNN. Selective Search performs the function of generating 2000 different regions that have the highest probability of containing an object. After we've come up with a set of region proposals, these proposals are then "warped" into an image size that can be fed into a trained CNN that extracts a feature vector for each region. This vector is then used as the input to a set of linear SVMs that are trained for each class and output a classification. The vector also gets fed into a bounding box regressor to obtain the most accurate coordinates.



R-CNN workflow

Mask – RCNN

Mask R-CNN is a Convolutional Neural Network (CNN) and state-of-the-art in terms of image segmentation. This variant of a Deep Neural Network detects objects in an image and generates a high-quality segmentation mask for each instance.

Mask R-CNN was built using Faster R-CNN. While Faster R-CNN has 2 outputs for each candidate object, a class label and a bounding-box offset, Mask R-CNN is the addition of a third branch that outputs the object mask. The additional mask output is distinct from the class and box outputs, requiring the extraction of a much finer spatial layout of an object.

Mask R-CNN is an extension of Faster R-CNN and works by adding a branch for predicting an object mask (Region of Interest) in parallel with the existing branch for bounding box recognition.

Training with Mask – RCNN

- First, we install, Matterport's Mask-RCNN model
- Downloaded COCO datasets pre-trained weights
- Formatted Dicom Images to JPG and annotated them
- Split the data into training and validation sets
- Trained the model and monitored train and validation loss

We used the Google's Object Detection API from TensorFlow to train the model.

We used a pretrained FPN. FRCNN model with multi-task loss in Mask R-CNN. The loss function for any image was defined as:

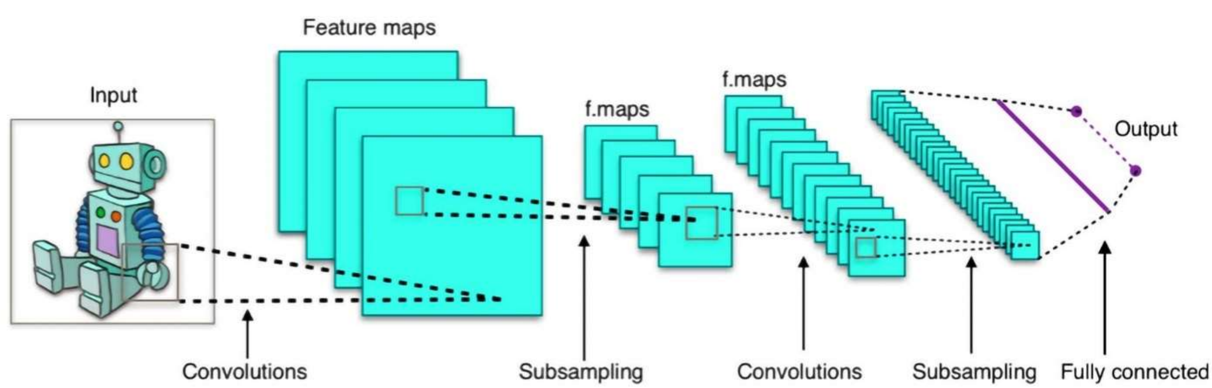
$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

We first used the CSV file and the images to make TFRecord files which can be fed to the object detection api. The TFRecord is an easy format for storing a sequence of binary records. As we are using large datasets, using a binary file format for storage of the data will have a great impact on the performance of the pipeline and will largely influence the training time.

3.2 Training with YOLOv5

YOLOv5 model

The YOLOv5 implementation has been done in Pytorch in contrast with the previous developments that used the DarkNet framework. This makes it easier to understand, train with it and deploy this model. There is no paper released with YOLO-v5. Architecturally it is quite similar to YOLO-v4. One difference is the use of Cross Stage Partial Network (CSP) to reduce computation cost.



The release of YOLOv5 includes five different models' sizes: YOLOv5s (smallest), YOLOv5m, YOLOv5l, YOLOv5x (largest).

YOLOv5 is one of the most recognized and popular object detectors. We use PyTorch based ultralytics implementation to train the images on YOLOv5. We compile the pytorch repository with GPU, CUDNN and OPENCV in Colab. Pre-trained weights were used, and we can set the parameters like learning rate, momentum, and weight decay according to our preference.
The weights are be stored after every specific interval of iterations in the backup directory. The loss function for YOLOv5 is:

$$L_{g_x,g_y} = z^T Q_{obj} z + \lambda_{noobj} \hat{c}_2^2 + p \circ log\hat{p} - (1-p) \circ log(1-\hat{p})$$

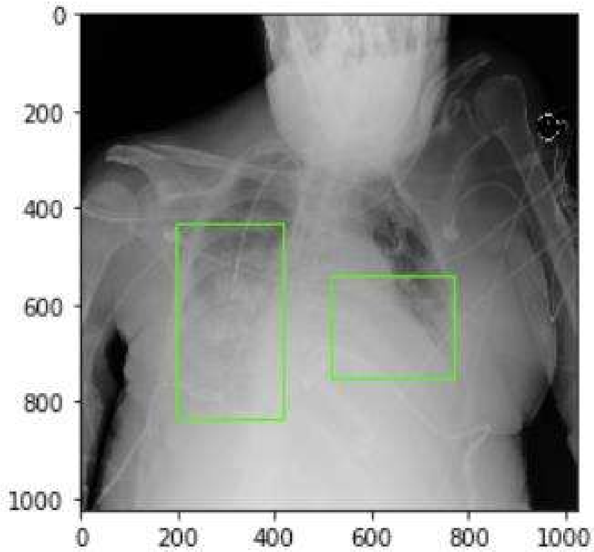
<u>Training with YOLOv5</u>

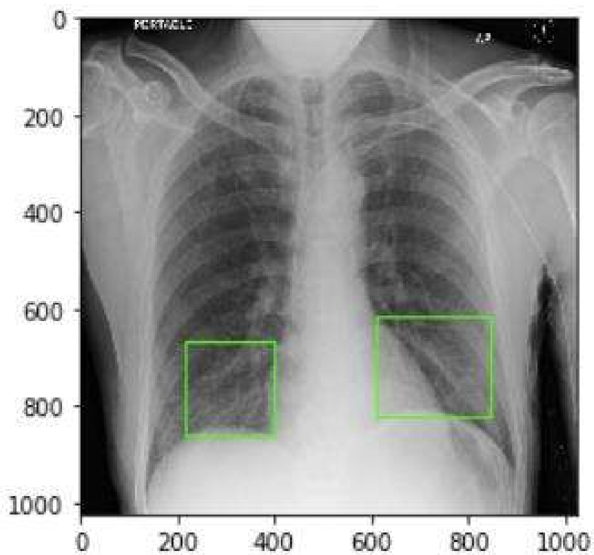
- First, we install, Ultralytics implementation of YOLOv5
- Downloaded COCO datasets pre-trained weights
- Formatted Dicom Images to JPG and annotated them
- Split the data into training and validation sets
- Trained the model and monitored train and validation loss on tensorboard

Results

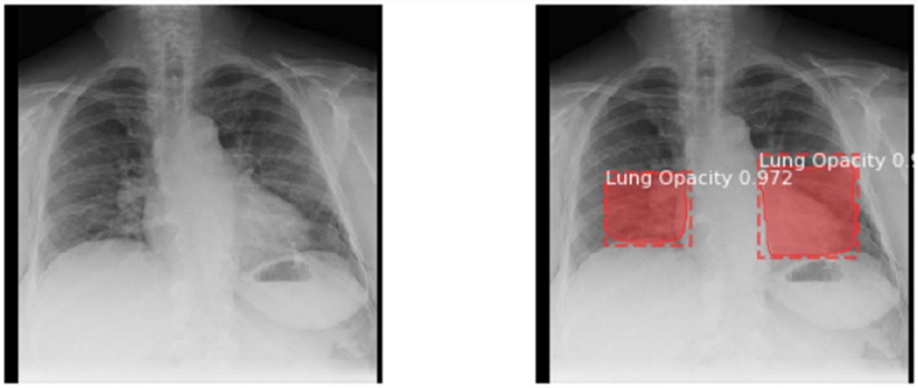
Viewing Results predicted by Mask R-CNN

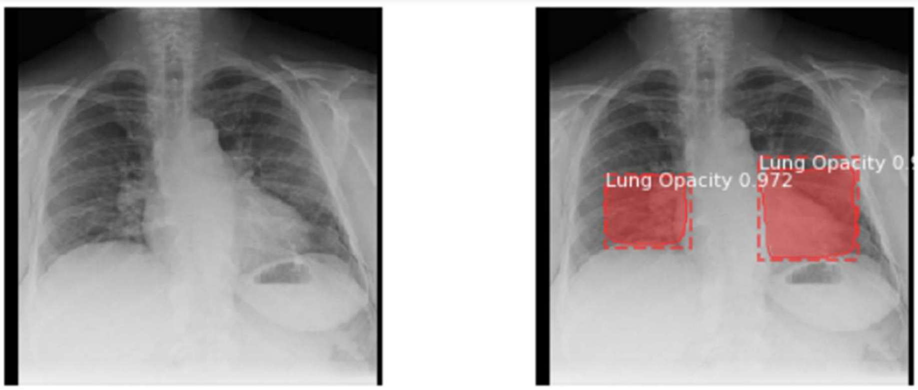
Here we can see CXR without pneumonia correctly predicted by the model.



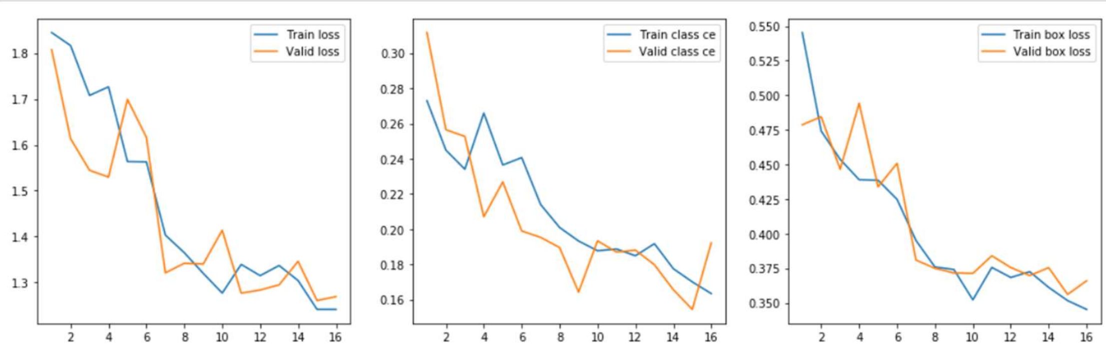


Here we see images of infected CXR correctly predicted by Mask-RCNN





The training and validation performance of Mask-RCNN over 16 epochs can be seen below:

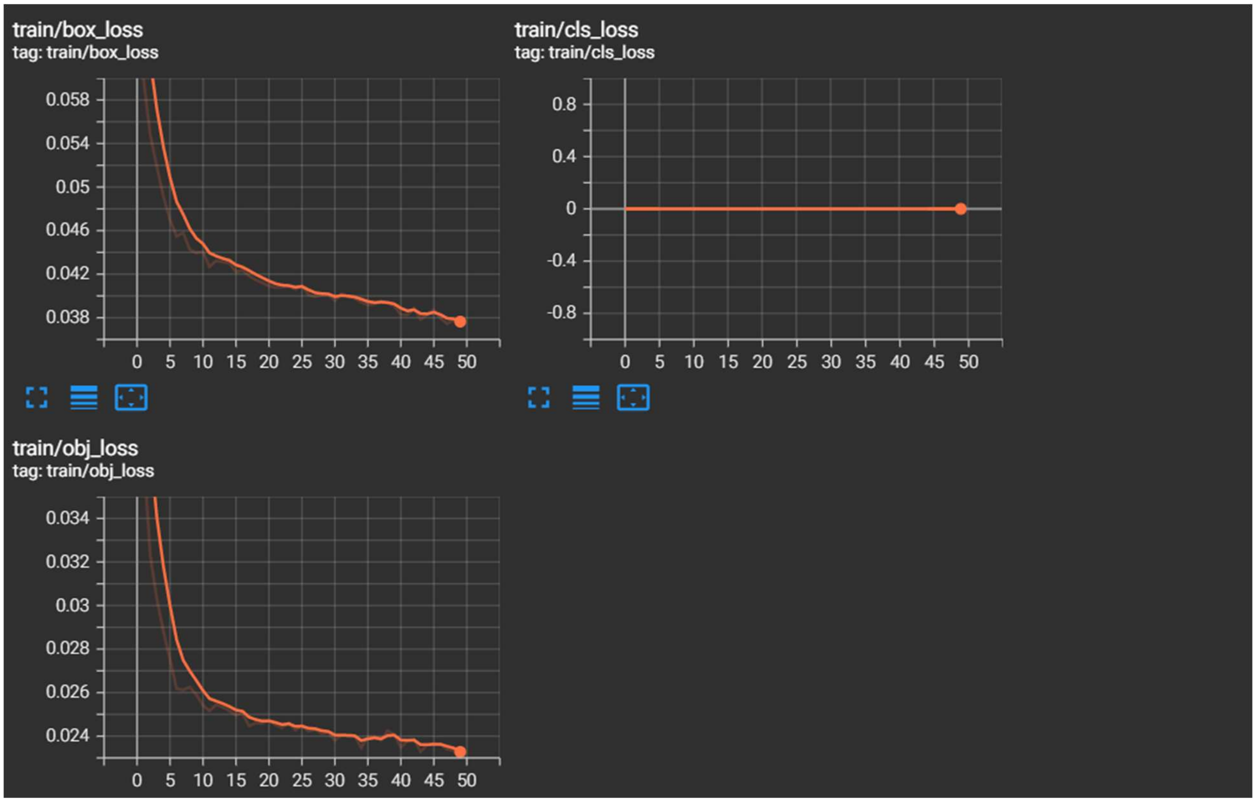


In [28]:
```
best_epoch = np.argmin(history["val_loss"])
print("Best Epoch:", best_epoch + 1, history["val_loss"][best_epoch])
```

```
Best Epoch: 15 1.2604877185821532
```

Viewing Results predicted by YOLO v5:

The training and validation performance of YOLOv5 over 50 epochs can be seen below:



Conclusion

In this work, we have successfully trained two models namely Mask - RCNN and YOLO3 and deployed a Pneumonia Detection model. After successfully training them and testing them, we have come to conclusion that Mask-RCNN is faster to train and test than YOLOv5. Loss of YOLOv5 for validation on object is 0.02 and loss of and Box/Loss is at 0.03 at 29th epoch respectively. Loss for Mask – RCNN is 1.2.

References

1. Schweitzer, D., & Agrawal, R. (2018). Multi-Class Object Detection from Aerial Images Using Mask R-CNN. *2018 IEEE International Conference on Big Data (Big Data)*. https://doi.org/10.1109/bigdata.2018.8622536
2. Armaanpriyadarshan/Training-a-Custom-TensorFlow-2.X . - GitHub. Retrieved May 6, 2022, from https://github.com/armaanpriyadarshan/Training-a-Custom-TensorFlow-2.X-Object-Detector
3. YOLOv3-RSNA Starting Notebook | Kaggle. Retrieved May 6, 2022, from https://www.kaggle.com/seohyeondeok/yolov3-rsna-starting-notebook

Link to Notebook

YOLOv5 Implementation

- https://colab.research.google.com/drive/1FAGQjw8fe6i0kYMmHv6F4qsLAeFLu0m7?usp=sharing

Mask – RCNN Implementation

- https://colab.research.google.com/drive/1ObjwaagrA7fyXwxsRZqwNTHASV4yamBF?usp=sharing