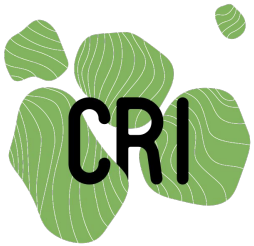
RSNA Pneumonia Detection Challenge

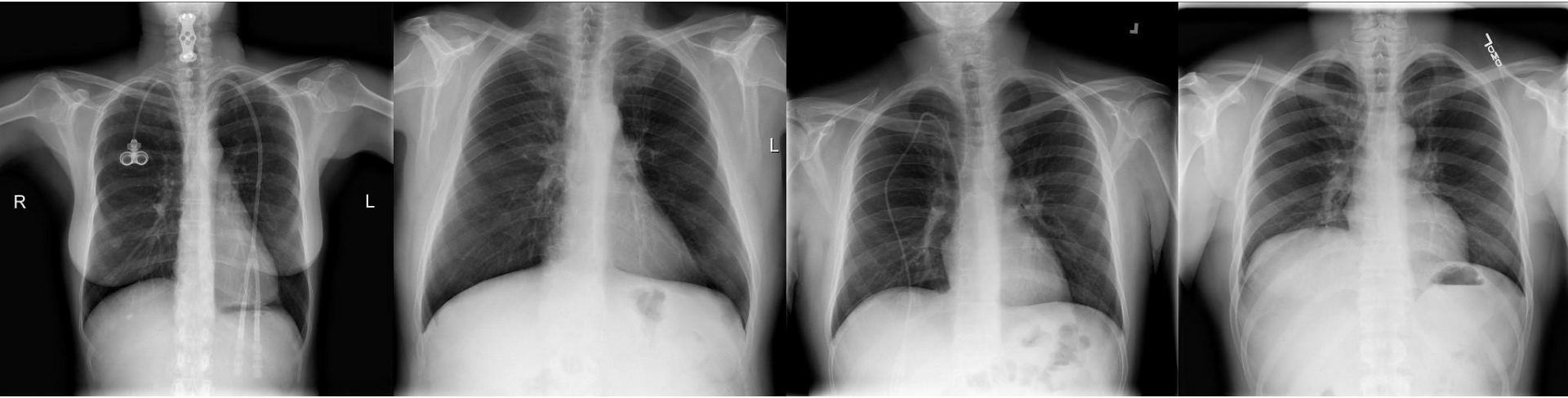


Presented by Fatima Wehbi, Marion Ficher, Qinghe Zeng and Sowmya Rajan.

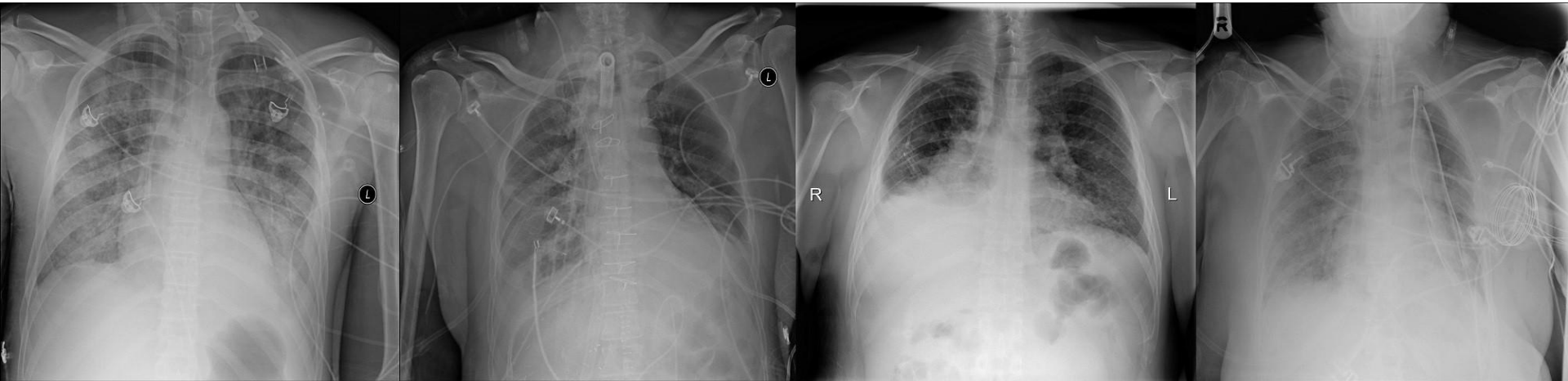
Outline

1. About Pneumonia
2. About the Challenge
3. Why AI?
4. Why YOLO?
5. Data Description
6. Methodology
7. Results
8. Conclusion

Chest X-ray of normal lungs



Chest X-ray of lungs with Pneumonia



About the challenge

RSNA : Radiology Society of North America

Society for Thoracic Radiology

MD.ai

Goal: To help develop artificial intelligence (AI) tools for radiology.

Why use AI?

- To accurately confirm a pneumonia diagnosis significantly faster than current clinical practice.

- It helps to accurately confirm the key findings leading to pneumonia.

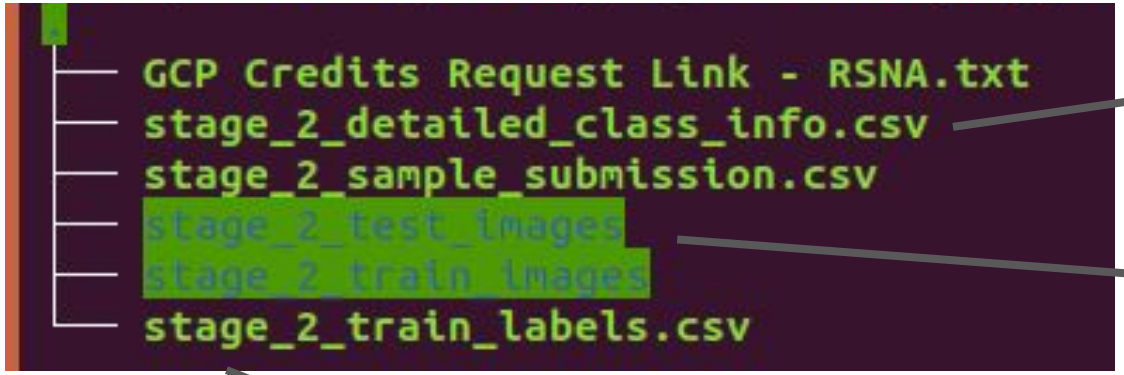
- To automate the image screening of potential pneumonia cases in order to prioritize and expedite their review.

Why YOLO?

When compared to YOLOv3, other algorithms such as Mask R-CNN, UNet, FCN, etc.

- Contain semantic segmentation tasks which are very slow and require more GPU resources.
- They require redundant parameter tuning and post processes.
- Difficulties occur during training.

Data Description



Patient ID	Class

PatientID.dcm
Training set
Test set

patientId	x	y	width	height	Target

Data Description

1 - Load datasets

```
# detailed_class_info
class_by_img = pd.read_csv('./data/stage_2_detailed_class_info.csv', sep=',')
# sample_submission
patients_testset = pd.read_csv('./data/stage_2_sample_submission.csv', sep=',')
# train_labels
patients_boudingboxes = pd.read_csv('./data/stage_2_train_labels.csv', sep=',')
```

```
print("class_by_img shape : ",class_by_img.shape)
print("patients_testset shape : ",patients_testset.shape)
print("patients_boudingboxes shape : ",patients_boudingboxes.shape)
```

```
class_by_img shape :  (30227, 2)
patients_testset shape :  (3000, 2)
patients_boudingboxes shape :  (30227, 6)
```

```
c_class = Counter(class_by_img['class'])
c_class
```

```
Counter({'No Lung Opacity / Not Normal': 11821,
         'Normal': 8851,
         'Lung Opacity': 9555})
```

```
c_targ = Counter(patients_boudingboxes['Target'])
c_targ
```

```
Counter({0: 20672, 1: 9555})
```

Methodology

Clone and Build YOLO v3 - test

Prepare the dataset to use it with YOLO

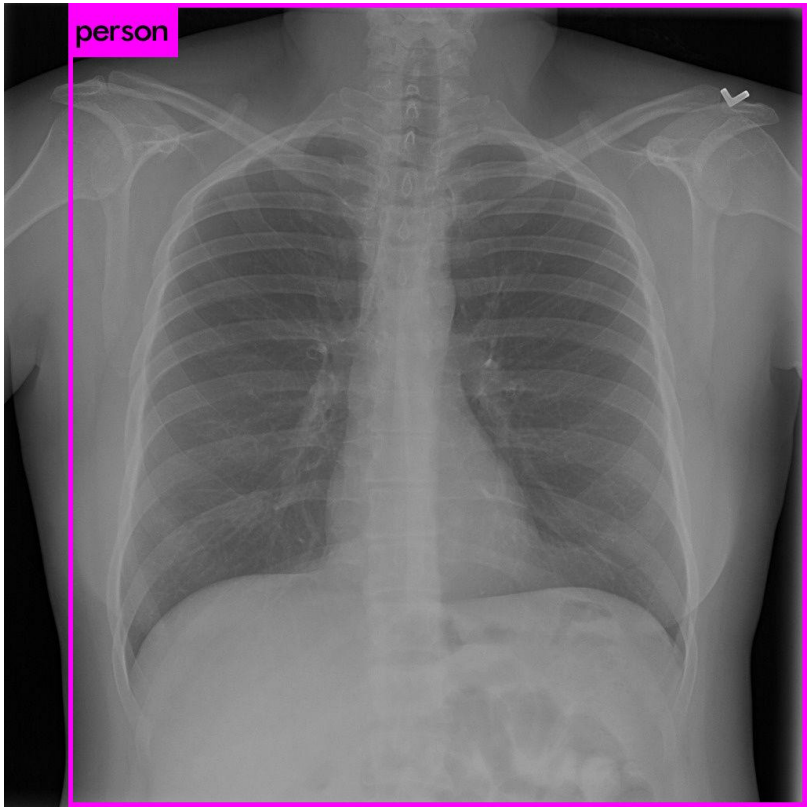
Divide into Training set and validation set

Clone and Build YOLO v3 - test

Loading weights from yolov3.weights...Done!

data/test.jpg: Predicted in 30.066087 seconds.

person: 93%

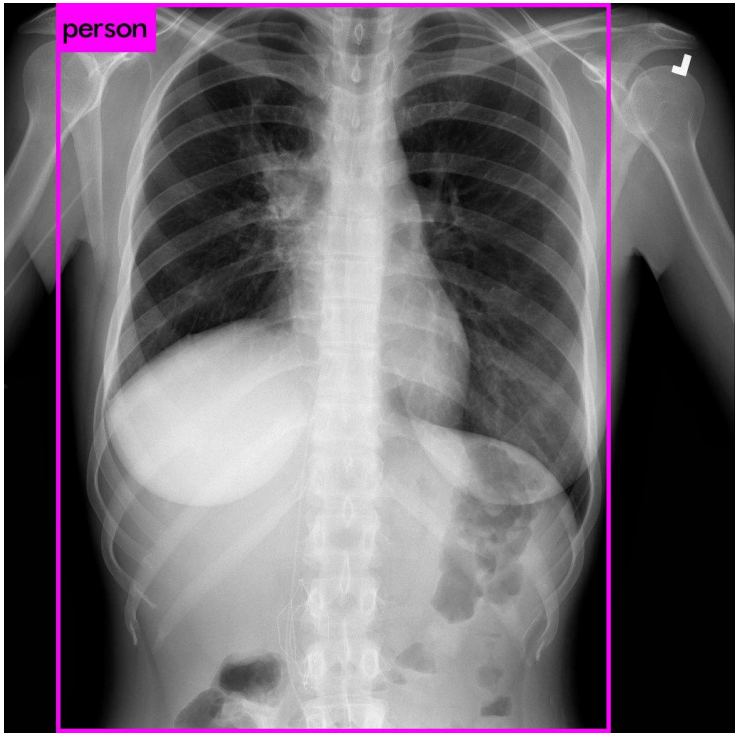


Clone and Build YOLO v3 - test

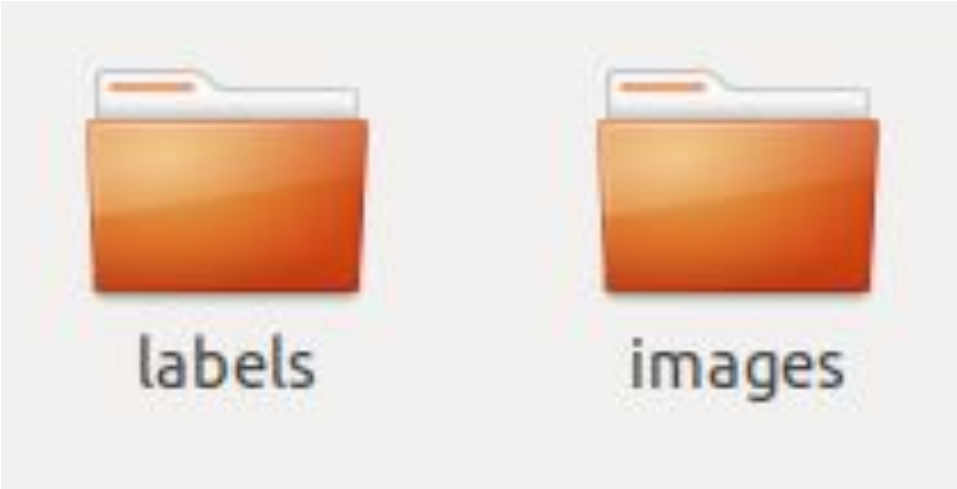
Loading weights from yolov3.weights...Done!

data/test_02.jpg: Predicted in 29.841259 seconds.

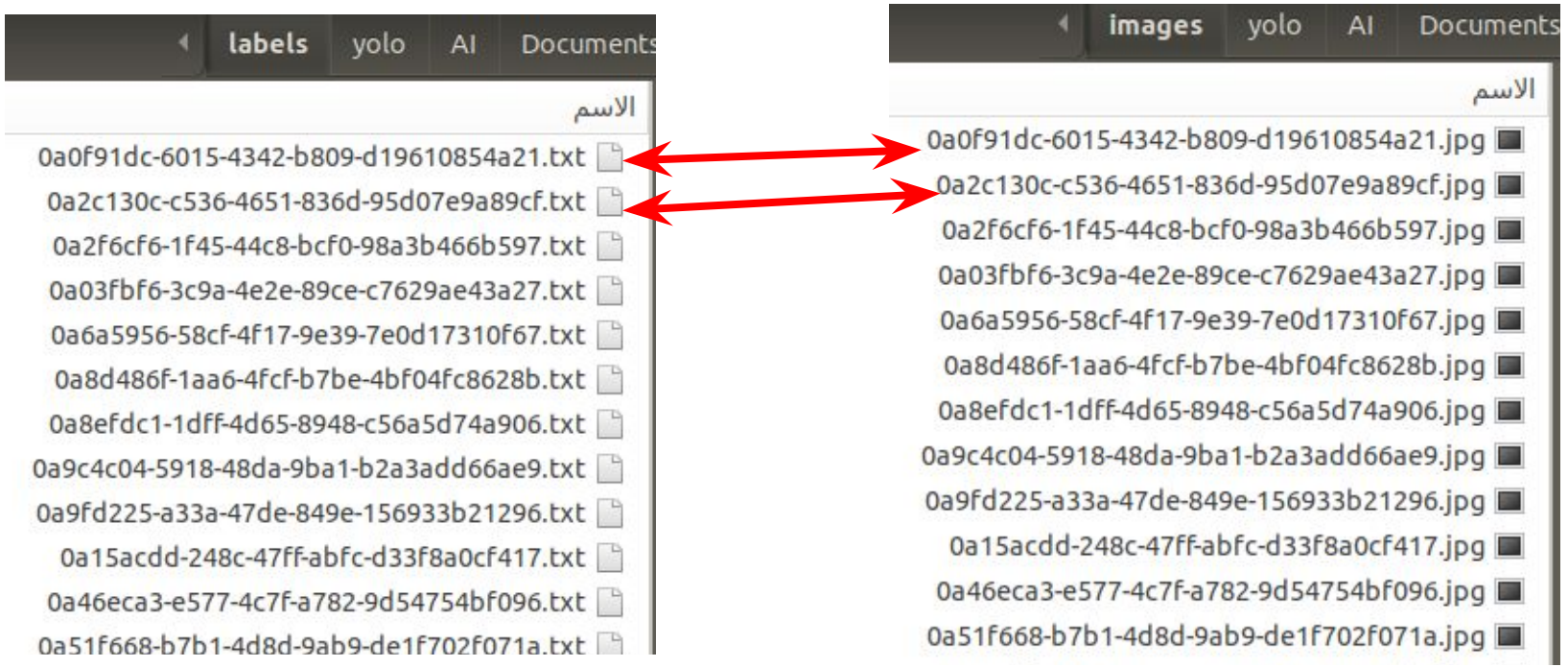
person: 66%



Preparing the dataset to use it with YOLO



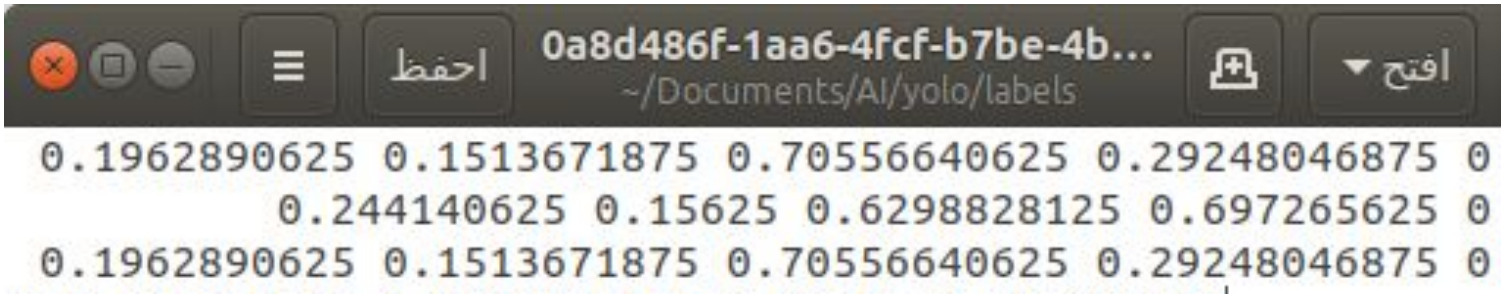
Preparing the dataset to use it with YOLO



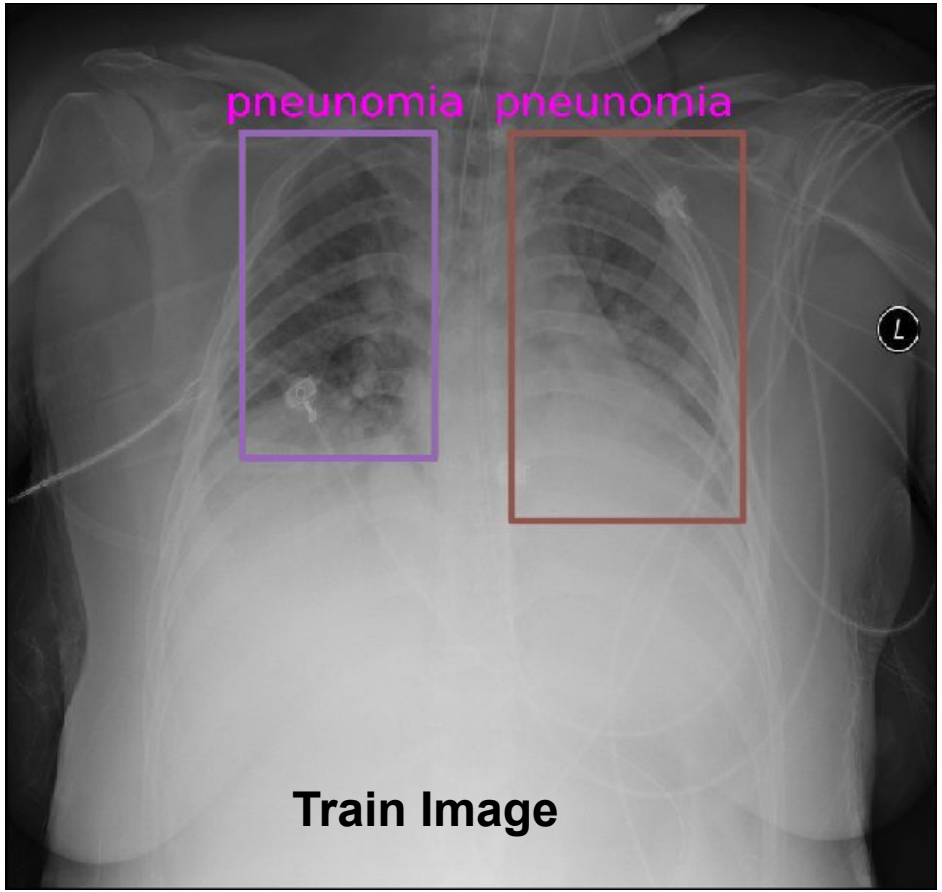
Convert images from .dcm to .jpg

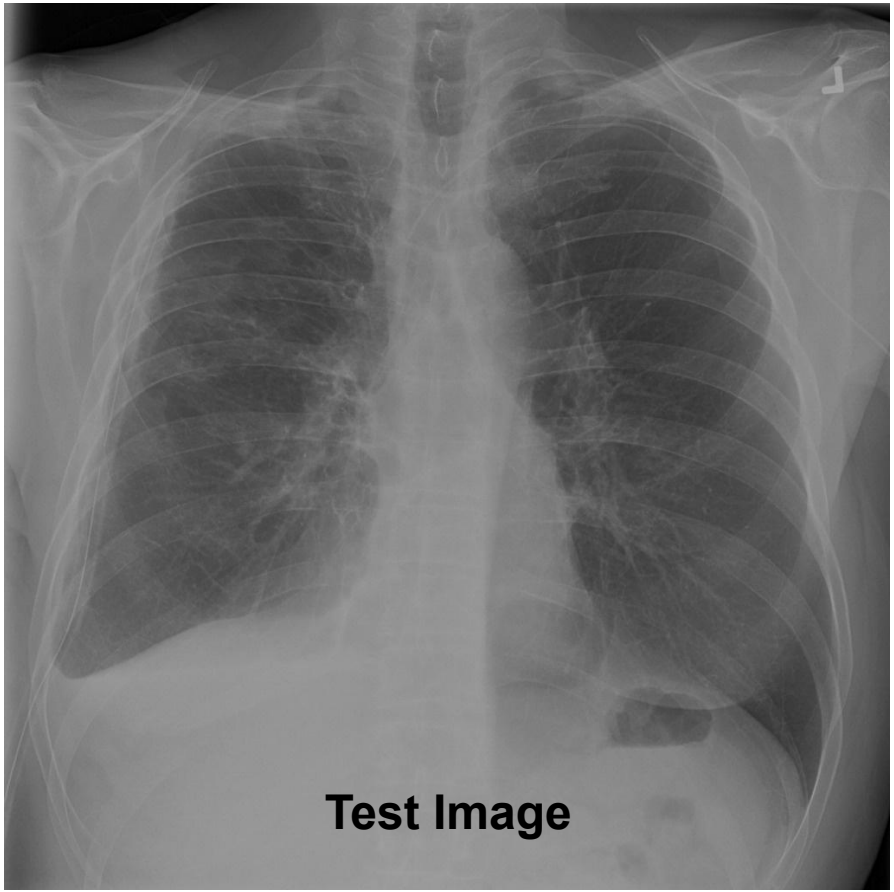
.Txt file

<object-class_1> <x_1> <y_1> <width_1> <height_1>



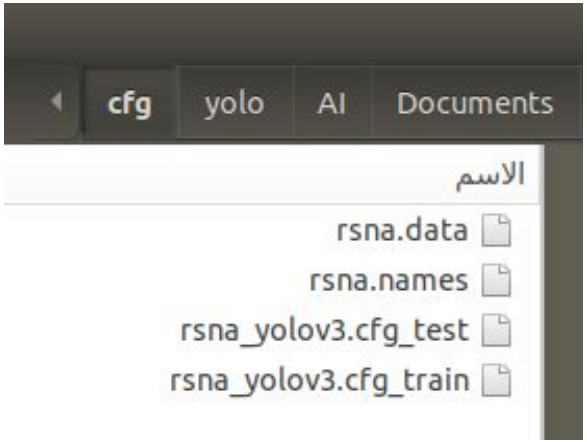
Example of training data and test data





Preparing YOLOv2 configuration files

A .txt file containing the paths of images to train YOLO with.



The text files like `2007_train.txt` list the image files for that year and image set. Darknet needs one text file with all of the images you want to train on. In this example, let's train with everything except the 2007 test set so that we can test our model. Run:

```
cat 2007_train.txt 2007_val.txt 2012_*.txt > train.txt
```

Now we have all the 2007 trainval and the 2012 trainval set in one big list. That's all we have to do for data setup!

Modify Cfg for Pascal Data

Now go to your Darknet directory. We have to change the `cfg/voc.data` config file to point to your data:

```
1 classes= 20
2 train  = <path-to-voc>/train.txt
3 valid  = <path-to-voc>2007_test.txt
4 names = data/voc.names
5 backup = backup
```

Weight file generated

Result - Modification of threshold