NTU Final Project 2019

HTC DeepQ

Outline

- Introduction
- Chest X-ray Pneumonia Detection
- Suggested Methods
- Rules of Competition
- Submission
- Evaluation Criteria
- Suggested Readings and Resources

Introduction

Prize

- Original
  - First prize: 10,000
  - Second prize: 5,000
  - Third prize: 2,000
- Current
  - 獲得特優之隊伍頒予獎狀乙紙，及新台幣一萬五千元獎金。
  - 獲得優等之隊伍頒予獎狀乙紙，及新台幣五千元獎金。
  - 獲得佳作之隊伍頒予獎狀乙紙，及新台幣一千元獎金。
  - 得獎作品隊伍之隊數及獎項由「宏達國際電子股份有限公司代表及授課老師」視各組隊數與作品優良情形議定，必要時可從缺、調整（並得將名額移至他題）或增加，但以不超過獎金總額為限，每題獎金總額為新台幣四萬元。
  - 15,000 x n1 + 5,000 x n2 + 1,000 x n3 <= 40,000

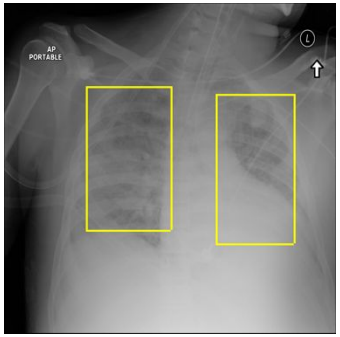
Contracts and non-disclosure agreement

- 立書人即_____(以下簡稱「乙方」)因利用宏達國際電子股份有限公司(以下簡稱「甲方」)提供之數據資料，參與甲方所舉辦之「胸腔X光影像肺炎偵測」科技競賽專案(以下簡稱「本競賽」)
- 乙方同意於其著作完成之同時無償讓與其上之著作財產權予甲方。
- 乙方同意負有保密義務。

Schedule

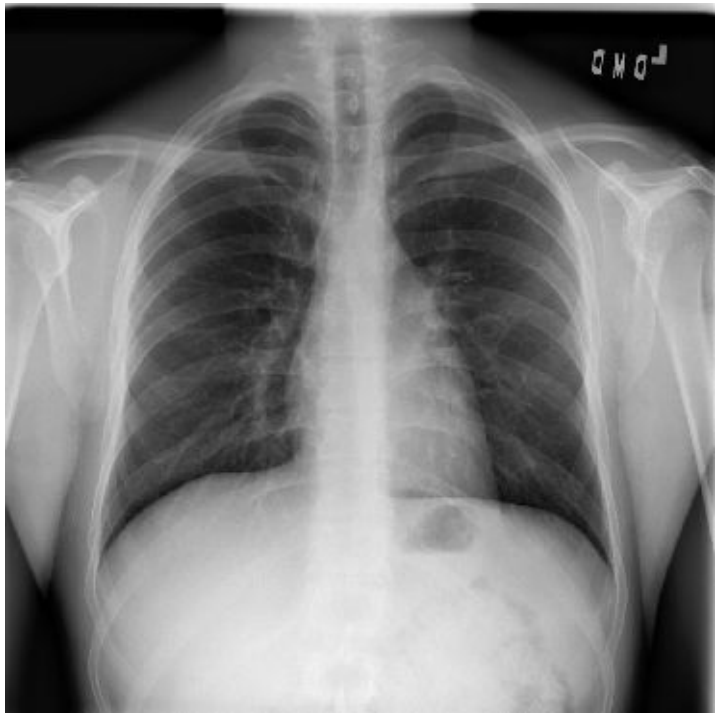
Date	TODO	Note (for NTU ML COURSE)
5/2 12:00:00 Thr.	Final Project Rules Announcement	
5/2 23:59:59 Thr.	Team Up Deadline	
5/30 23:59:59 Thr.	Proposal Deadline	Proposal + 1%
	Early Simple Baseline	Early Simple + 1%
6/13 23:59:59 Thr.	Early Strong Baseline	Early Strong + 1%
6/27 23:59:59 Thr.	Final Presentation	Bonus + 3%
	Contracts and non-disclosure agreement	
6/30 23:59:59 Sun.	Final Project Ranking Final Project Github Deadline (Report & Github)	Ranking (2%) Pass Simple + 2% Pass Strong + 2%

Chest X-ray Pneumonia Detection

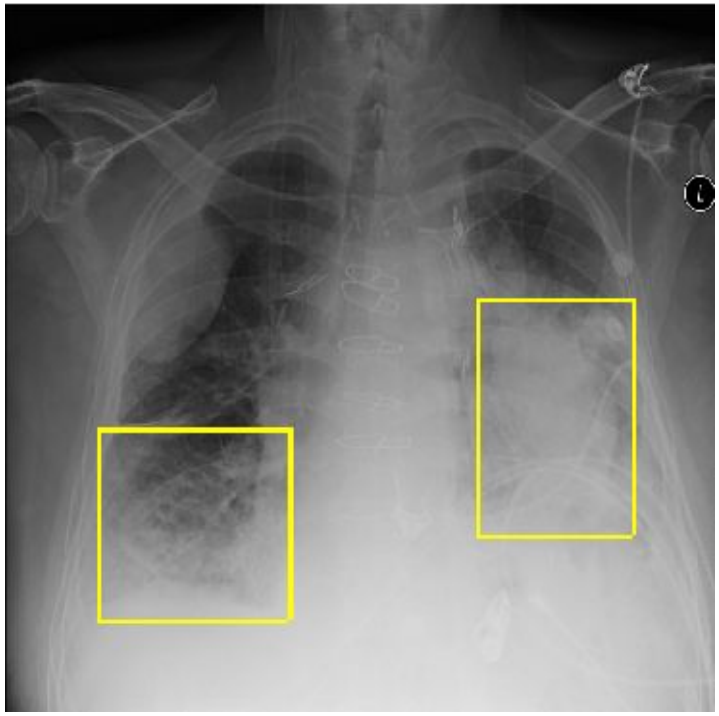


- Total Images : 26,788
  - train: 21765
  - test: 5023
- These images are from
  - RSNA Pneumonia Challenge
    - https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/
  - NIH Chest X-ray Dataset
    - https://www.kaggle.com/nih-chest-xrays/data

Examples

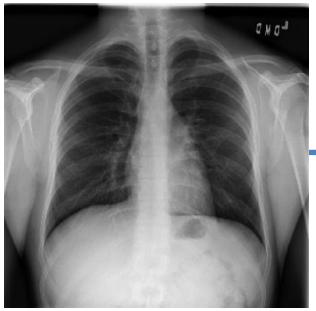


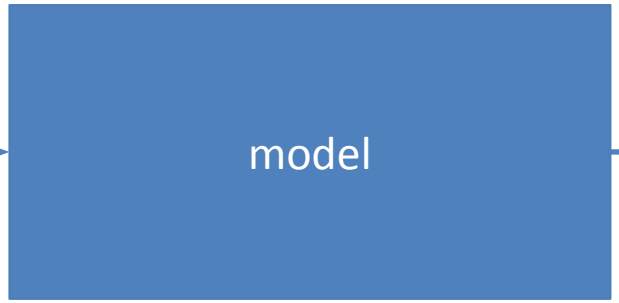
No Pneumonia



Pneumonia

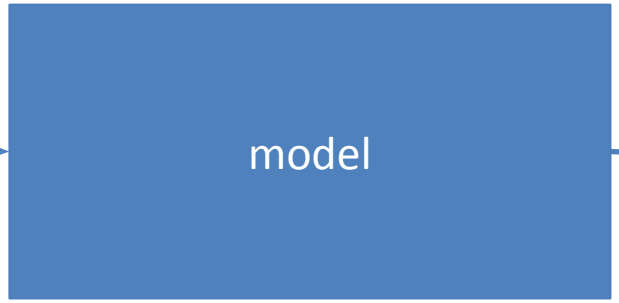
Your Model

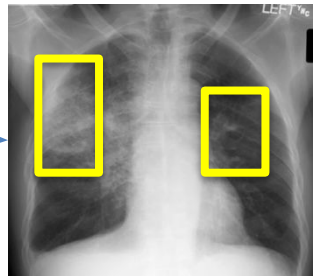




Input (No Pneumonia)

Output





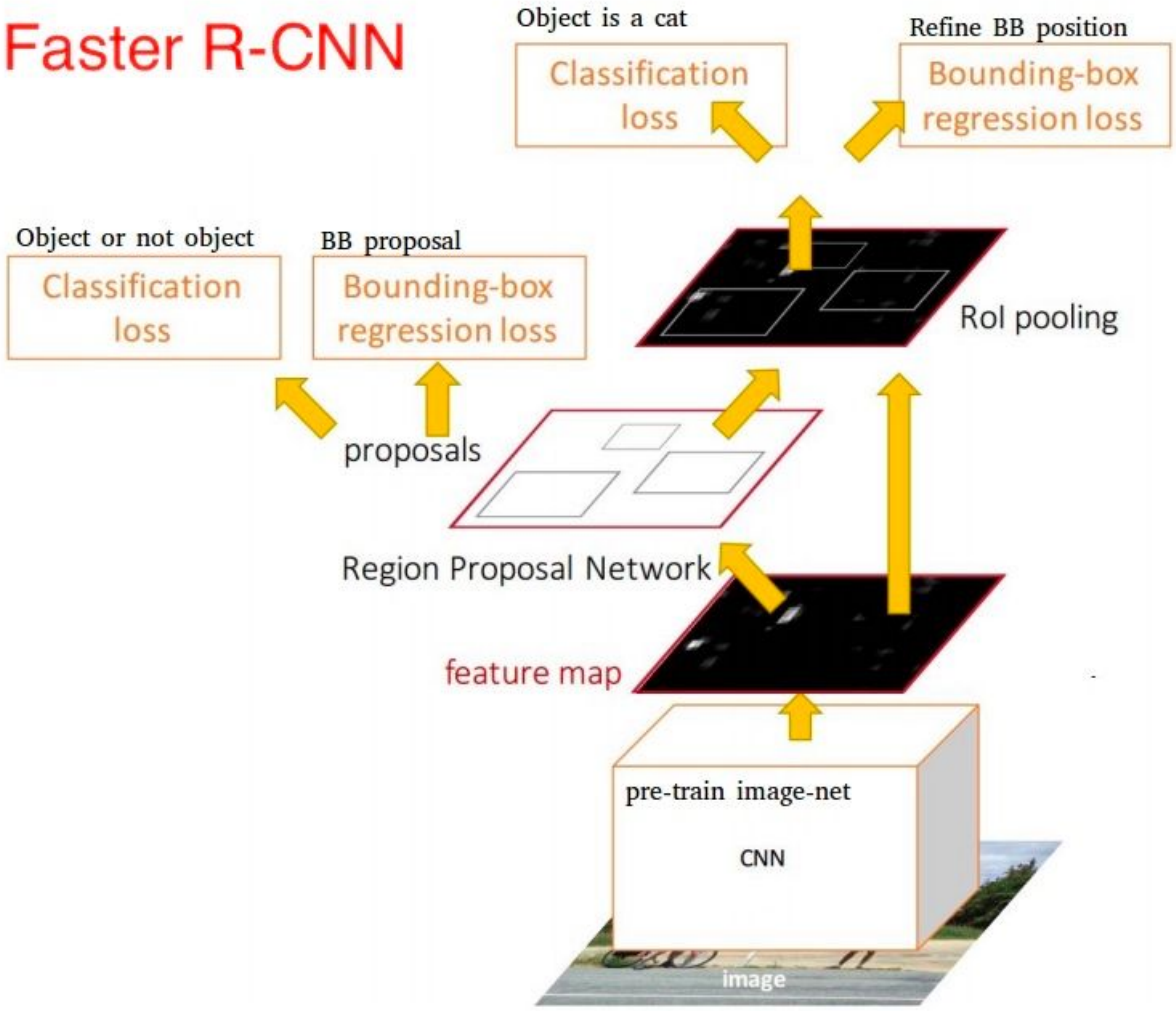
Input (neumonia)

Output

Suggested Methods

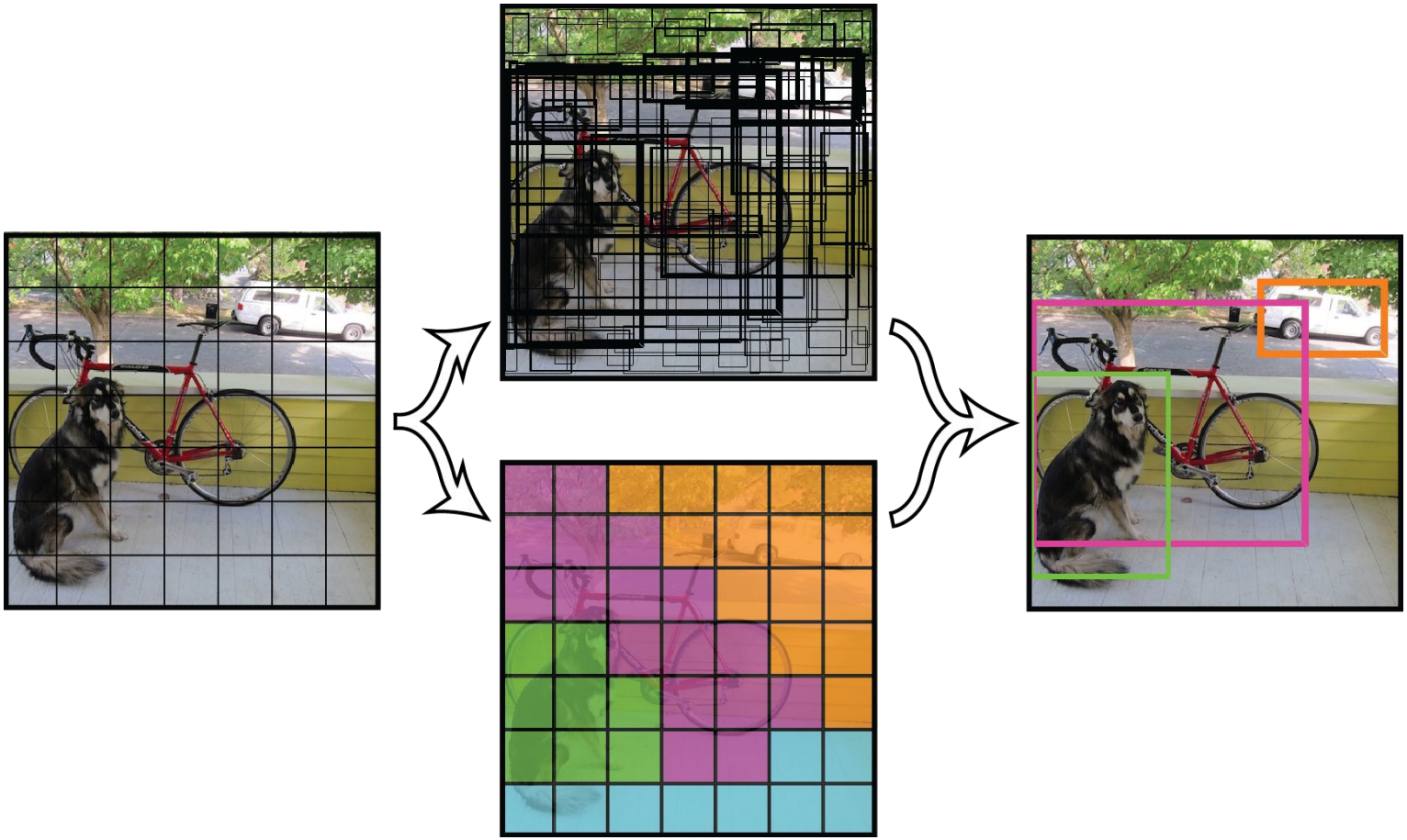
- Faster RCNN
- YOLO
- Transfer Learning
- Multitask Learning

Faster RCNN



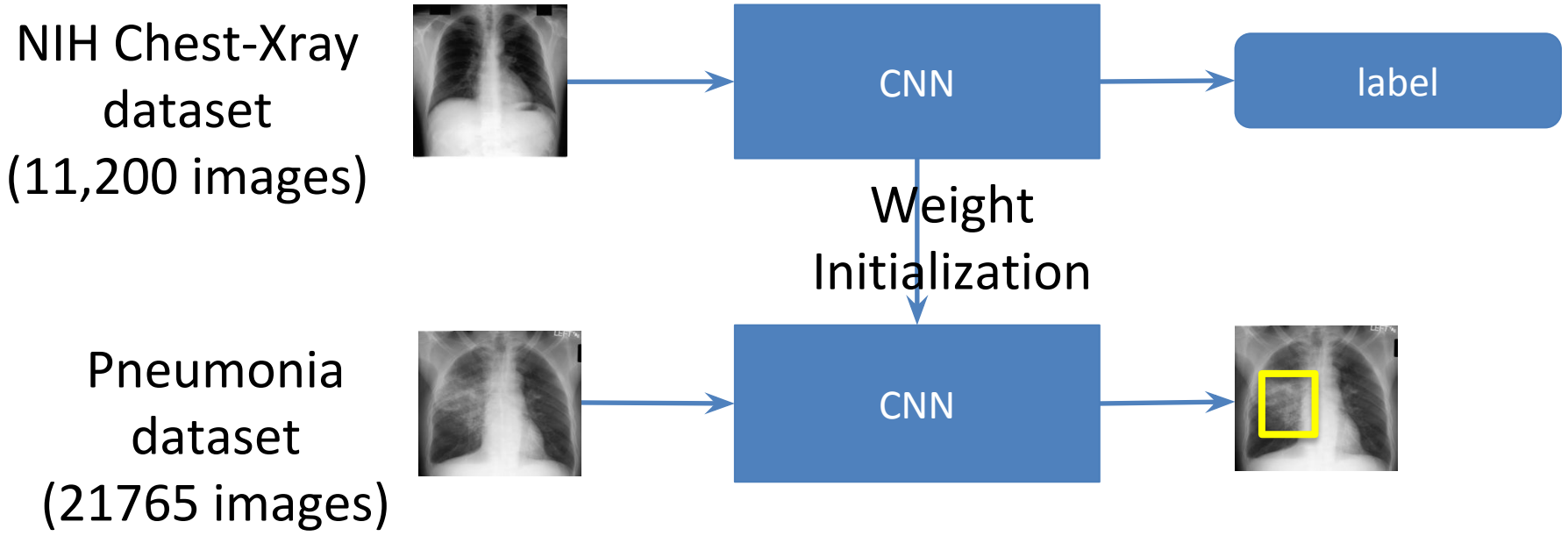
https://arxiv.org/abs/1506.01497

Yolo



https://arxiv.org/abs/1612.08242

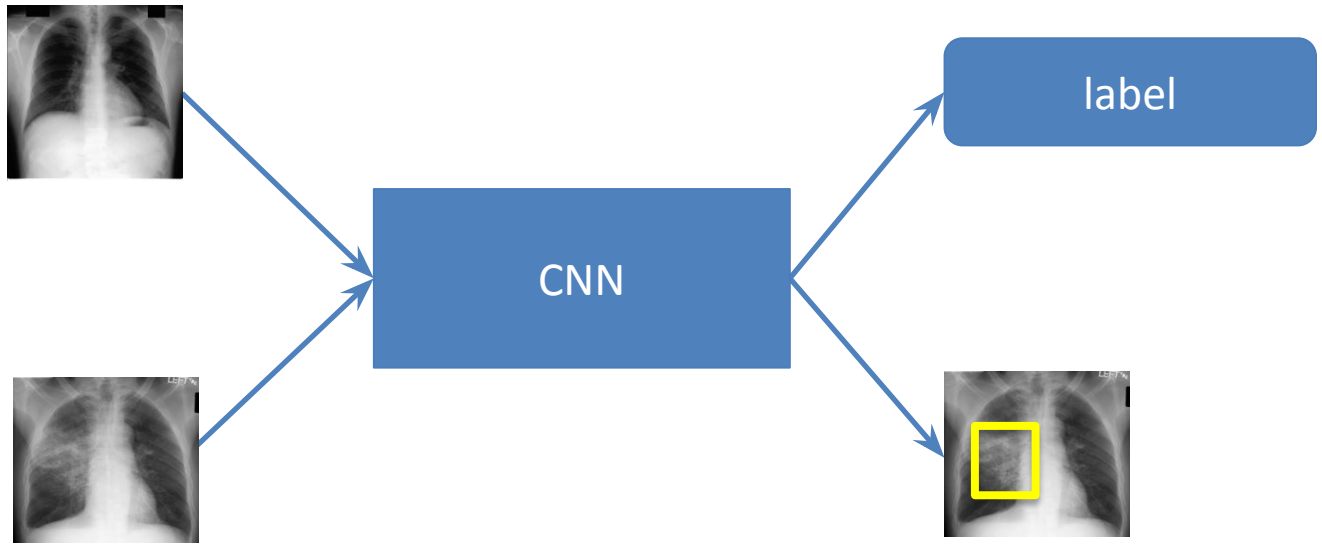
Transfer Learning



Multitask Learning

NIH Chest-Xray dataset (11,200 images)

Pneumonia dataset (21765 images)



Other novel Ideas are also welcome

Rules of Competition

- Training Data
- Pre-trained Model
- Code of Your Implementation

Training Data

- You are only allowed to use the images and labels and other meta provided by our kaggle competition website, **NIH-Chest X-ray dataset**, and **ImageNet dataset** to train your model.
- **Do not use RSNA Pneumonia Challenge Dataset** to train your model.
- Do not use any other images, labels or datasets to train your model.
- Data Augmentation (such as rotation, crop, and GAN) and Data Cleaning are allowed, but they should be run by script.
- Manually create additional labels or manually adjust existing labels is allowed, but you should post your results on the discussion forum of our kaggle competition website.

Pre-trained Model

- You are allowed to download any available pre-trained model from the Internet, but this pre-trained model should only be trained on **NIH-Chest X-ray dataset** or **ImageNet dataset** or both.

Code of Your Implementation

- Your algorithm should be implemented in **python** programming language.
- You are allowed to reuse any code available from the Internet. However, you should be aware of its license.

Submission

- Submission to kaggle website
  - During this competition, you can submit your model to kaggle website and evaluating it on testing data.
- Final submission
  - After you complete your project, you should submit your whole project so that we can verify your result.

Submission to kaggle website

- You can upload your results to our NTU Final Project Kaggle webpage to see your score on Public LeaderBoard.

Submission to kaggle website

1. **Train your model**
2. **Create prediction.csv**: Inference your model
3. **Create submission.csv**: Convert to file with RLE format.
4. **Upload submission.csv**

Step 1 - Train your model

- Download these files:

  ./train: all raw train images

  ./test: all raw test images

  train_labels.csv: annotated ground truth (bounding box format).

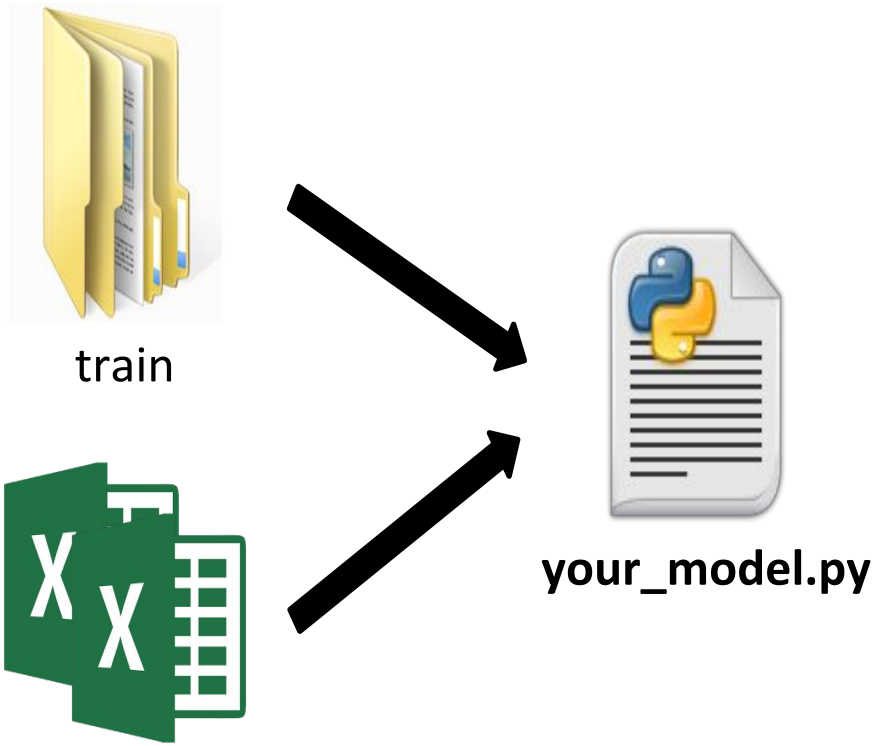
  train_metadata.csv: list of patient's attributes (sex, view, age) to help you develop your model.

  test_metadata.csv: list of patient's attributes (sex, view, age) to help you make inference of your model.

  bbox_to_rle.py: script to convert b-box to RLE format

Step 1 - Train your model

- Your model should be implemented in Python.

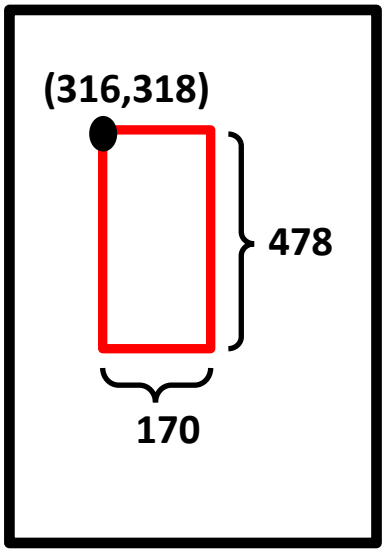


Step 1 - Train your model

- **train_labels.csv**

000db696-cf54-4385-b10b-6b16fbb3f985 , 316 , 318 , 170 , 478 , 1

Image ID X Y width height Target



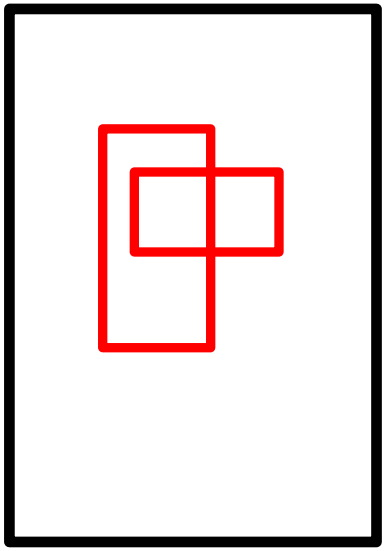
./train/000db696-cf54-4385-b10b-6b16fbb3f985.png

Step 1 - Train your model

- Multiple bounding boxes

  000db696-cf54-4385-b10b-6b16fbb3f985 , 316 , 318 , 170 , 478 , 1

  000db696-cf54-4385-b10b-6b16fbb3f985 , 427 , 428 , 300 , 100 , 1

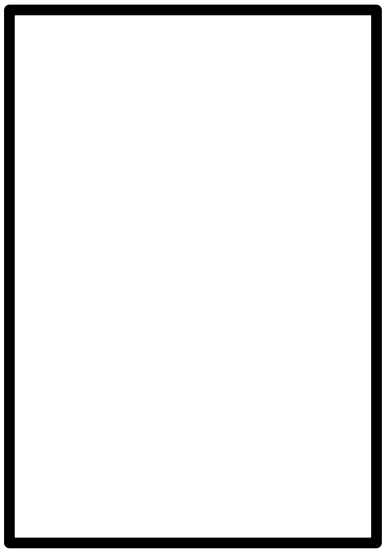


./train/000db696-cf54-4385-b10b-6b16fbb3f985.png

Step 1 - Train your model

- No bounding box

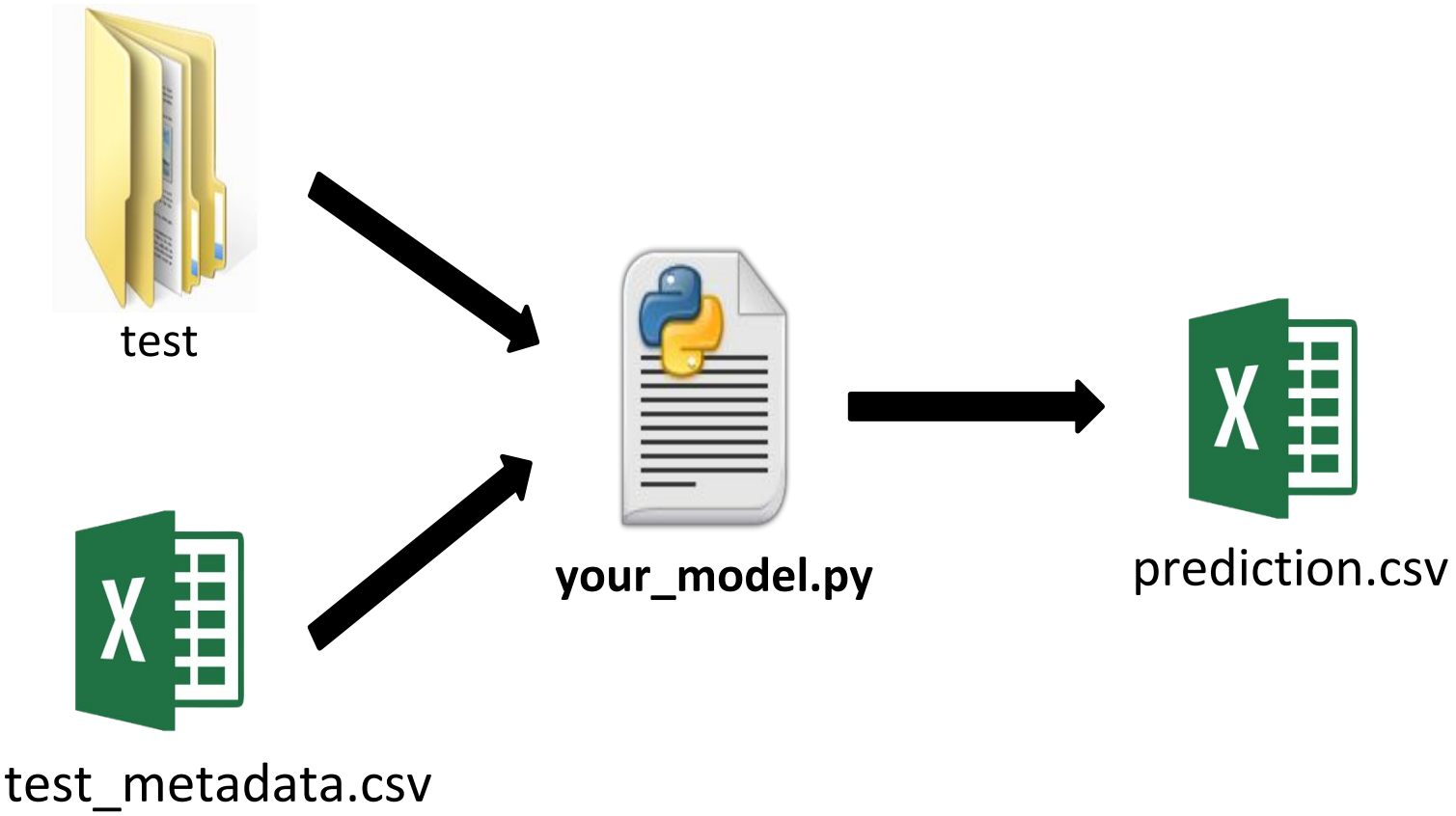
000db696-cf54-4385-b10b-6b16fbb3f985 , , , , , 0



./train/000db696-cf54-4385-b10b-6b16fbb3f985.png

Step 2 - Create **prediction.csv**

- **prediction.csv:** contains bounding box information.



Step 2 - Create **prediction.csv**

test

patient0000.png
patient0001.png
…

prediction.csv

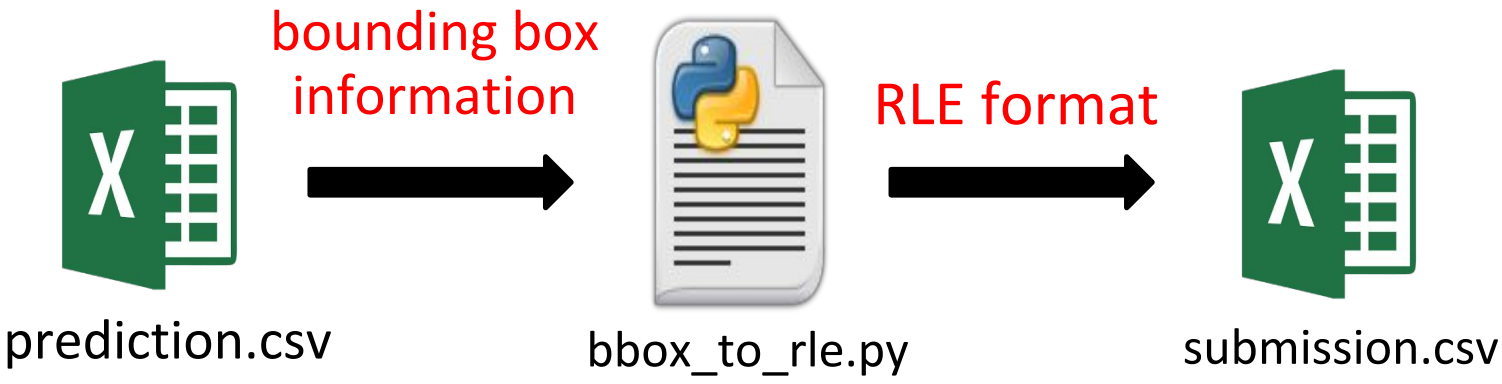
patientId	x	y	width	height	Target
patient0000.png	0	0	0	0	0
patient0001.png	100	200	300	400	1

filename

bounding box information

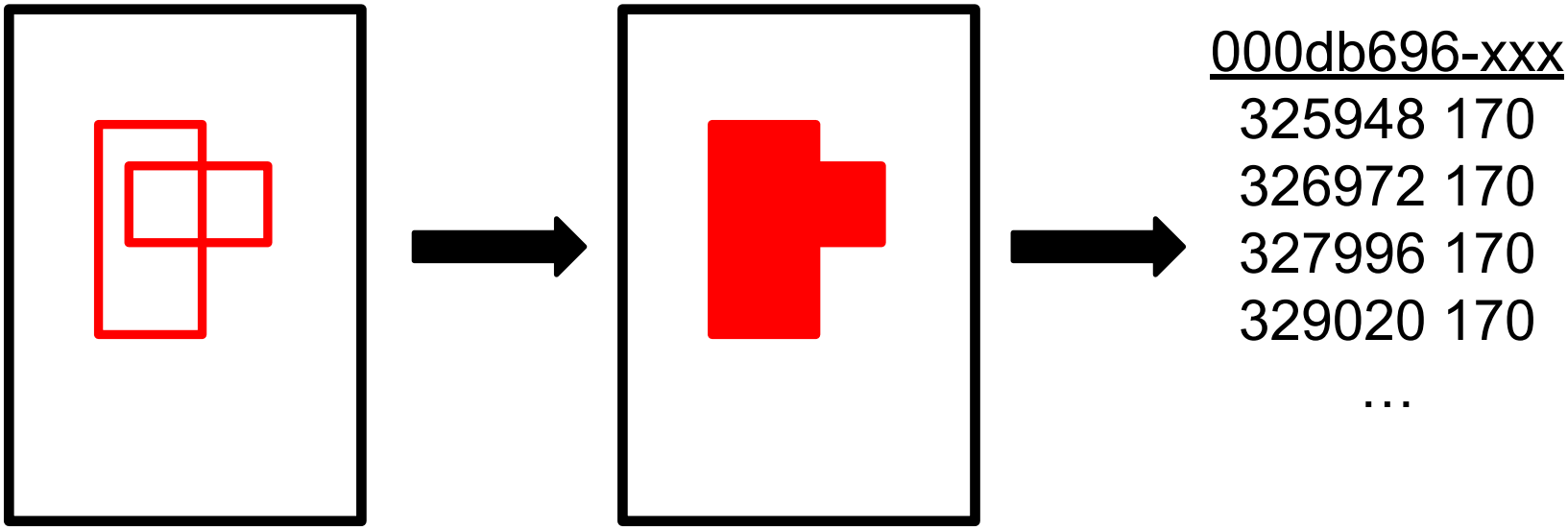
Step 3 - Create **submission.csv**

- Kaggle InClass competition only supports submission file with run-length encoding (RLE) format.
- **bbox_to_rle.py** has already been implemented.



Step 3 - Create **submission.csv**

- **bbox_to_rle.py**
  1. Compute union of bounding boxes.
  2. Encode into **pixel-wise** RLE format.



000db696-cf54-4385-b10b-6b16fbb3f985 , 316 , 318 , 170 , 478 , 1
000db696-cf54-4385-b10b-6b16fbb3f985 , 427 , 428 , 300 , 100 , 1

Step 3 - Create **submission.csv**

prediction.csv

patientId	x	y	width	height	Target
patient0000.png	0	0	0	0	0
patient0001.png	100	200	300	400	1

submission.csv

patientId	EncodedString
patient0000.png	1 1
patient0001.png	204900 300 205294 300 …..

filename

RLE format

Step 4 - Upload **submission.csv**

- Simply submit your **submission.csv** to Kaggle website.

Final Submission

- You should submit your code to a github repository
- This repository should contain the following:
  - /src: Code that can reproduce your result from scratch
  - report.pdf
  - README.md
  - requirements.txt
- You should not submit the dataset or trained model to this github repository
- If you use any pre-trained model, you should provide a link in README file to download the pre-trained model.

Code to submit

- You should should submit all the code that can reproduce your final result from scratch.
- **If we fail to reproduce your result, we may consider you cheating, and you may be disqualified in this competition.**

Format of README file

- You should submit a README file of your implementation.
- The file is about **how to run your code and reproduce your result. If we follow this file, your result should be able to reproduce.**
- This file should contain the following:
  - Where to download your dataset or pre-trained model.
  - How to set up the environment to run your code.
  - What is your final result, and how to run your code to reproduce your result.

Format of Report

- Besides README file, you should submit a report (report.pdf) to introduce the theoretical and experimental parts of your project.
- Your report should contain at least (but not restricted to) the following sections:
  - Introduction & Motivation
  - Data Preprocessing/Feature Engineering
  - Methods (At least two different methods)
  - Experiment and Discussion
  - Conclusion
  - Reference
- Your report should be written in Chinese, and contains 6 to 12 pages.

Evaluation Criteria for DeepQ Competition

- Score:
  - 80% : kaggle private leaderboard dataset.
  - 20% : Report , README file
- Prizes:
  - The teams whose **score are higher than public leaderboard strong baseline** are eligible to win the prizes.

Suggested Readings

- Faster R-CNN
  - https://arxiv.org/abs/1506.01497
- YOLO2
  - https://arxiv.org/abs/1612.08242

Suggested Readings

- Transfer Learning
  - How transferable are features in deep neural networks?
    - https://arxiv.org/abs/1411.1792
  - Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?
    - https://arxiv.org/abs/1706.00712
  - What makes ImageNet good for transfer learning?
    - https://arxiv.org/abs/1608.08614

Suggested Source Code for Chest X-Ray Dataset

- CheXNet implementation in PyTorch
  - https://github.com/zoogzog/chexnet
- X-Net: Classifying Chest X-Rays Using Deep Learning
  - https://github.com/gregwchase/nih-chest-xray

Suggested method for Pneumonia Detection

- YOLO
  - https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolo v2-28b1b93e2088
  - https://github.com/xiongzihua/pytorch-YOLO-v1
  - https://github.com/vietnguyen91/Yolo-v2-pytorch
  - https://github.com/ayooshkathuria/pytorch-yolo-v3
- RCNN
  - https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e
  - https://github.com/multimodallearning/pytorch-mask-rcnn
  - https://github.com/jwyang/faster-rcnn.pytorch
- RetinaNet
  - https://towardsdatascience.com/review-retinanet-focal-loss-object-detection-38fba6afabe4
  - https://github.com/yhenon/pytorch-retinanet
  - https://github.com/kuangliu/pytorch-retinanet

Suggested method for Pneumonia Detection

- RSNA Pneumonia Challenge 1st place solution:
  - https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/discussion/70421#latest-496413
- RSNA Pneumonia Challenge 2nd place solution:
  - https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/discussion/70427#latest-497399
- RSNA Pneumonia Challenge 3rd place solution:
  - https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/discussion/70632#latest-440310

If you have any question about the rules, the dataset, or any other question about this competition, please send an email to Mark Chang mark.fc_chang@htc.com .

If you have any other question about this course, please send an email to NTU ML TA ntumlta2019@gmail.com, thanks.