Digital Medicine 2021 Case Presentation2: Pneumonia / Covid Classification using CXR

Team: CT_06_CCC
310551056 Cheng-Ju Ho
310551118 Wei-Cheng Chien
310551165 Yen-An Chien

Our Github Project Link:
https://github.com/luluho1208/NYCU-Digital-Medicine-2021-HW2

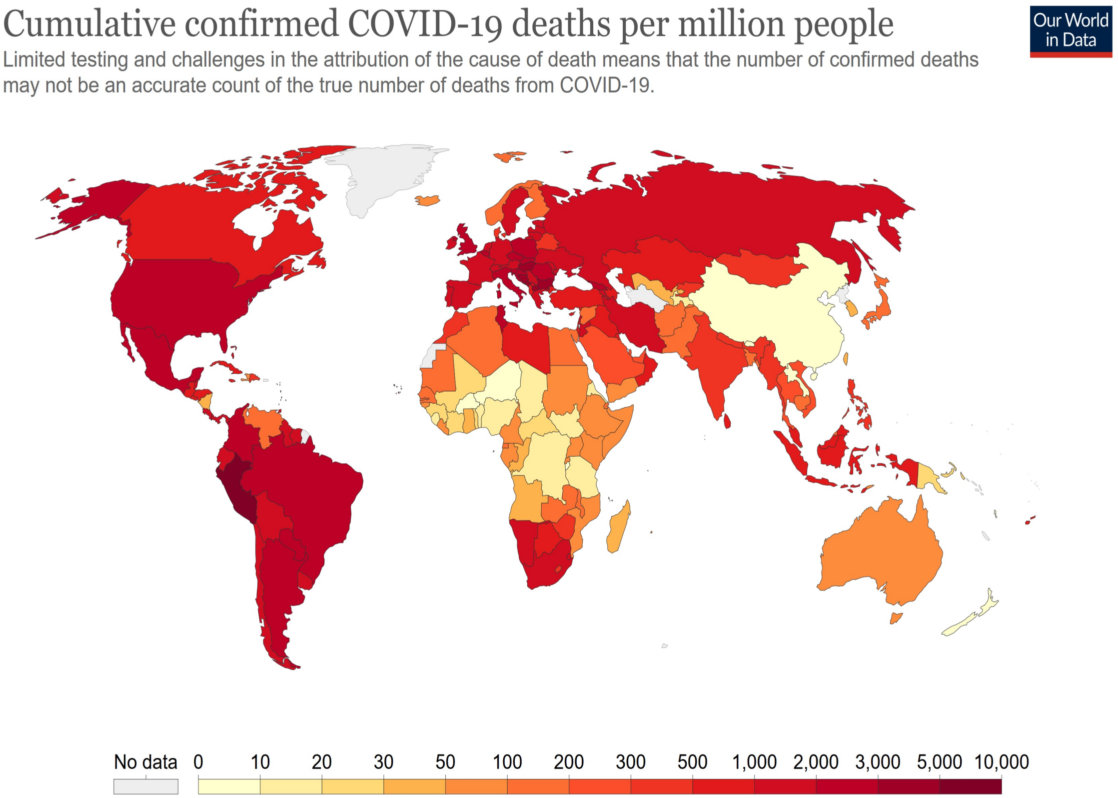
Outline

01.

Introduction

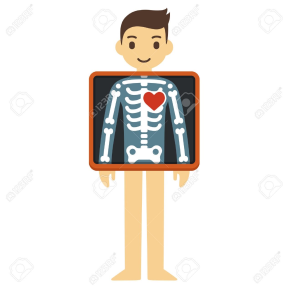
Team: CT_06_CCC

Covid-19

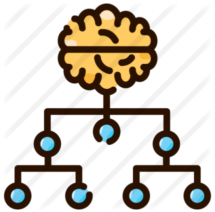


Team: CT_06_CCC

Introduction

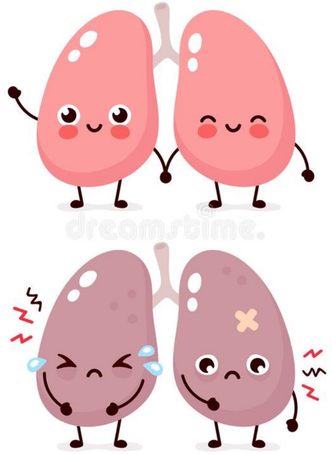


Patient's CXR



Classifier

Healthy ☺



Covid ☹

MISSION

Use given CXRs to tell whether the patients are Confirmed or not.

VISION

Provide the automatic pipeline to help our hospital system.

Our Dataset

Kaggle: SIIM-FISABIO-RSNA COVID-19 Detection

Featured Code Competition

SIIM-FISABIO-RSNA COVID-19 Detection

Identify and localize COVID-19 abnormalities on chest radiographs

$100,000
Prize Money

Society for Imaging Informatics in Medicine (SIIM) · 1,305 teams · 3 months ago

Overview Data Code Discussion Leaderboard Rules Team My Submissions **Late Submission** ···

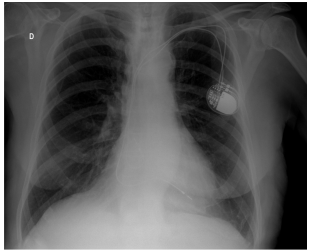
Overview

- Description
- Evaluation
- Timeline
- Prizes
- Code Requirements
- Call For Models
- Acknowledgments
- Partners: HP & Intel

Five times more deadly than the flu, COVID-19 causes significant morbidity and mortality. Like other pneumonias, pulmonary infection with COVID-19 results in inflammation and fluid in the lungs. COVID-19 looks very similar to other viral and bacterial pneumonias on chest radiographs, which makes it difficult to diagnose. Your computer vision model to detect and localize COVID-19 would help doctors provide a quick and confident diagnosis. As a result, patients could get the right treatment before the most severe effects of the virus take hold.

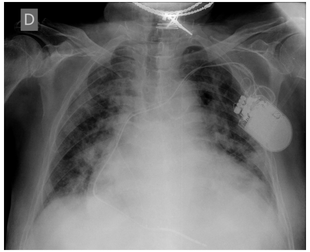
Currently, COVID-19 can be diagnosed via polymerase chain reaction to detect genetic material from the virus or chest radiograph. However, it can take a few hours and sometimes days before the molecular test results are back. By contrast, chest radiographs can be obtained in minutes. While guidelines exist to help radiologists differentiate COVID-19 from other types of infection, their assessments vary. In addition, non-radiologists could be supported with better localization of the disease, such as with a visual bounding box.

Our Data Label



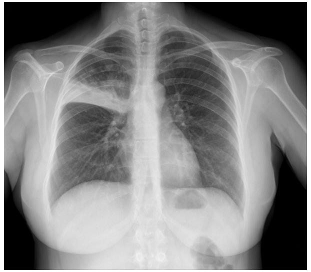
Negative (normal)

400 images for training



Typical (pneumonia)

400 images for training



Atypical (covid-19)

400 images for training

02.

Our Method

Method Keyword

Data Augmentation
& Enhancement

Vision Transformer

Transfer Learning

Cross-Validation &
Model Ensemble

Our pipeline

Step1 | Step2 | Step3 | Step4 | Step5

Step1: Data Pre-Processing

- Read dicom file
- apply_voi_lut
- Normalization
- Resize to 1024
- Save png file

Step2: Data Augmentation

- RandomAffine (rotation, scale)
- RandomCrop
- RandomHorizontalFlip
- HistEq
- Normalize

Step3: Model Training

- Resnet-50
- Resnet-101
- Vision Transformer

Step4: Prediction and Result

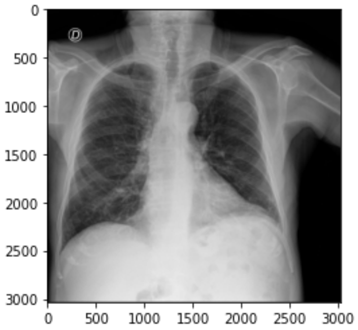
- Ensemble 5 ViT models for prediction.

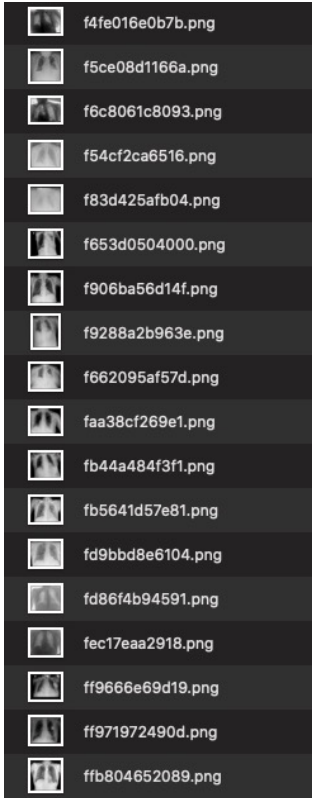
Step5: Analyze

- Compare different hyperparameter
- Compare different data augmentation

Step1 : Data Pre-Processing

1. Read the dicom format data from Kaggle .
2. Do the normalization (0 ~ 255).
3. Resize the image to 1024 with ratio-fixed.
4. Convert to png file and save.





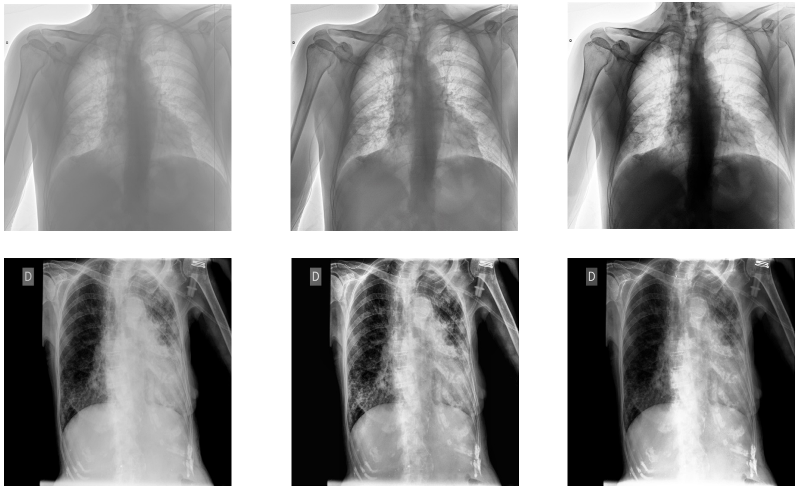
Team: CT_06_CCC

Step2 : Data Augmentation & Enhancement

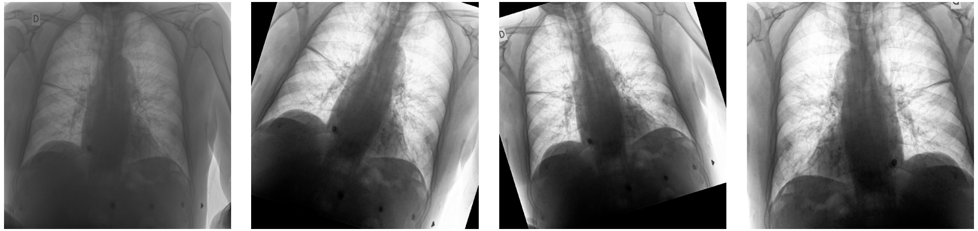
```
train_transform = transforms.Compose([
        transforms.Resize((int(528 * 1.10), int(528 * 1.10))),
        transforms.RandomAffine(degrees=10, scale=(0.9, 1.1)),
        transforms.RandomCrop((528, 528)),
        transforms.RandomHorizontalFlip(p=0.5),
        hisEqulColor(),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
    ])
```

- Image_size = 528
- AUG_TIMES = 3
- HistEq vs CLAHE (Contrast Limited adaptive HistEq)

Original HistEq CLAHE



ORI AUG1 AUG2 AUG3



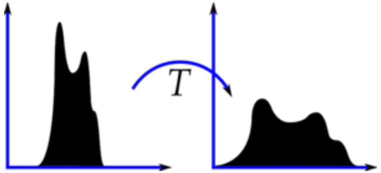
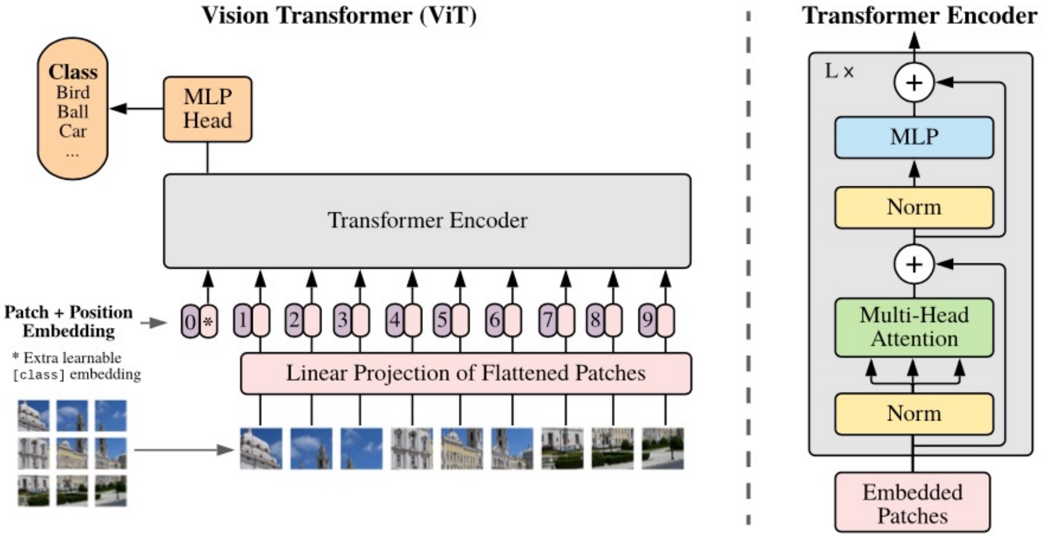


Fig :HistEq

Team: CT_06_CCC

Step3 : Model Training

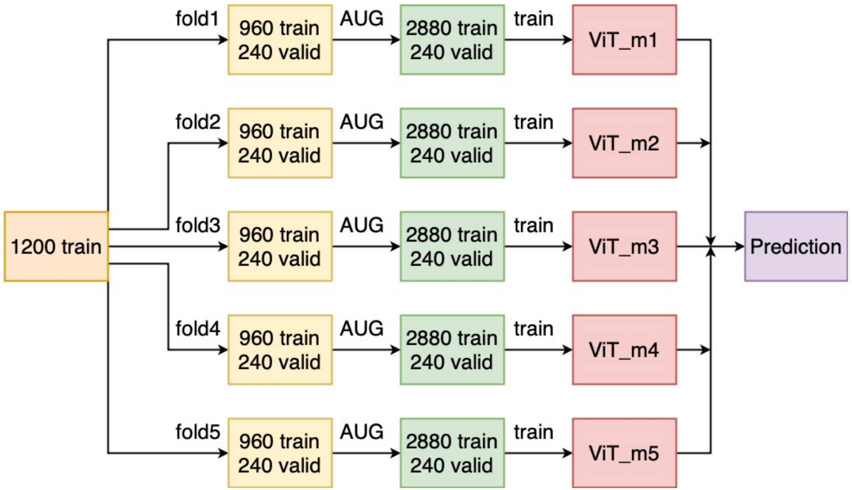
- Vision-Transformer (https://github.com/jeonsworld/ViT-pytorch)
- Google provide pretrained weights, which is powerful in image classification.
- Pretrained on imagenet21K. (Transfer Learning)



Team: CT_06_CCC

Step3 : Model Training

- We choose the Vision-Transformer to be our final model (ViT_B-16.npz).
- Split ratio of the training/validation is 8:2.
- Use 5-fold Cross-Validation to train 5 ViT model.
- To maintain the batchsize large enough, use 8 GeForce RTX 2080Ti for training.



Team: CT_06_CCC

Step3 : Model Training

Hyper-parameter

- Loss function: CrossEntropyLoss
- Optimizer: SGD(model.parameters(), momentum=0.9, weight_decay=1e-2)
- Scheduler: ReduceLROnPlateau(optimizer, factor=0.1, patience=3)

```
parser.add_argument('--kfold', type=int, default=0)
parser.add_argument('--batch_size', type=int, default=16)
parser.add_argument('--epoch', type=int, default=30)
parser.add_argument('--lr', type=float, default=0.001)
parser.add_argument('--image_size', type=int, default=528)
parser.add_argument('--aug_time', type=int, default=3)
parser.add_argument('--patience', type=int, default=3)
```

Step4 : Prediction and Result

- We use 5 models to predict the test CXR and vote for the final result.
- We achieve 0.57 mean f1-score on public leaderboard.
- We achieve ___ mean f1-score on private leaderboard.

Public Leaderboard　Private Leaderboard

This leaderboard is calculated with approximately 70% of the test data.
The final results will be based on the other 30%, so the final standings may be different.

Raw Data　Refresh

#	Team Name	Notebook	Team Members	Score	Entries	Last
1	CT_09_Overfitting			0.64761	9	19m
2	CT_05_AIGILE			0.63809	7	3d
3	CT_02_IKEA			0.60000	26	13h
4	CT_01_B@seLine			0.57142	58	3h
5	CT_06_CCC			0.57142	19	1h

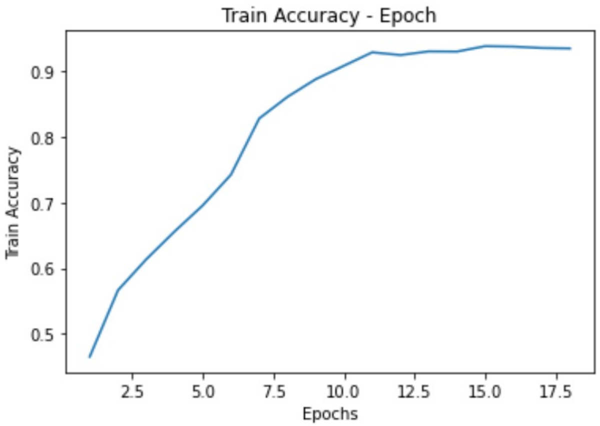
Step4 : Prediction and Result

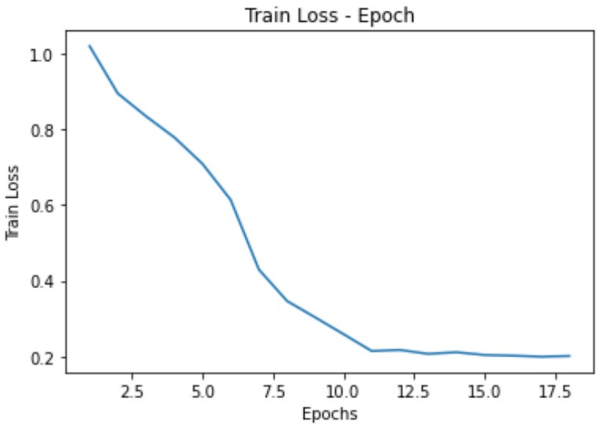
- We use 5 model to predict the test CXR and vote for the final result.

F1-score	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Public
Resnet50	0.58	0.61	0.59	0.62	0.59	0.514
Resnet101	0.6	0.6	0.63	0.58	0.59	0.523
ViT-B_16	0.64	0.65	0.64	0.64	0.62	0.571
ViT-B_32	0.63	0.65	0.62	0.64	0.63	0.552
ViT-L_16	0.67	0.67	0.62	0.64	0.63	0.552

Step4 : Prediction and Result

- Below is our training / validation processing.

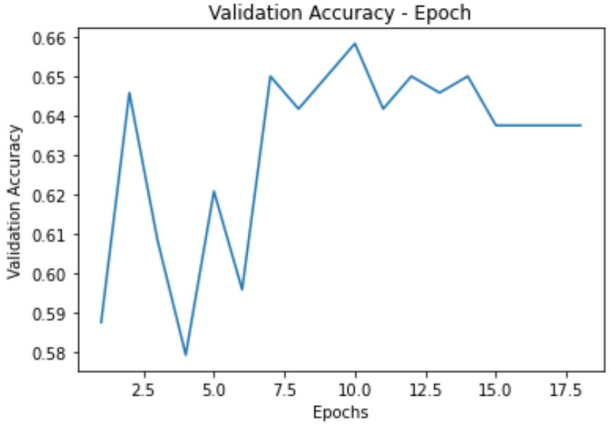


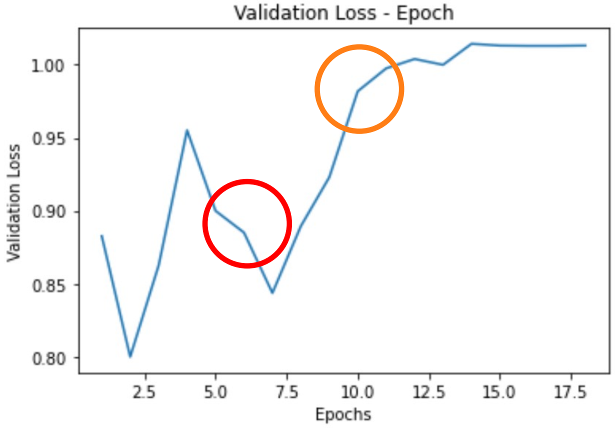


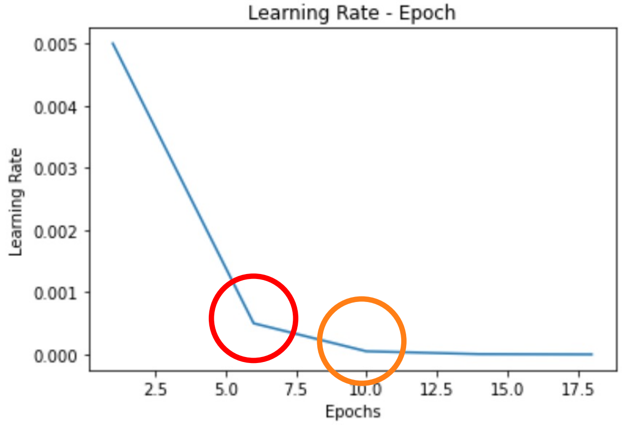
Team: CT_06_CCC

Step4 : Prediction and Result

- Below is our training / validation processing.







03.

Discussion

Discussion

- ViT is a very powerful model, but we only have 1200 training data => Overfitting

Data Augmentation

- If we use strong augment (rotate 30, scale 1.3x...), it will hurt model performance.
- If we augment the data to 5x, it will overfit the training data.

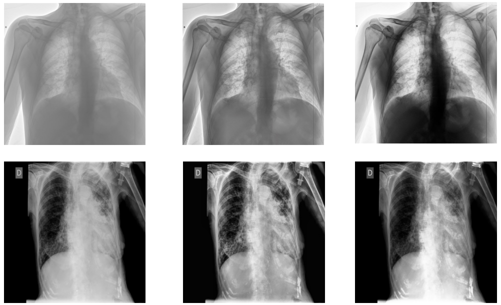
Model Training – Batch_Size

- If the batchsize is small, it tends to overfit. (BatchNorm)
- For 528 * 528 images, it takes lots of time for training.

Model Training – Learning rate & Scheduler

- Hard to training, so we need to choose lr, and scheduler carefully
- For such a Huge model, we need to have a larger lr to train, but it will also overfit quickly. We set lr to 0.001 and use ReduceLROnPlateau(factor=0.1, patience=3) to control the learning rate.

Discussion



Data Enhancement - HistEq vs CLAHE

- It is important to do the data enhancement.
- We use HistEq, but we find it has same effort to use HistEq or CLAHE.

Model Training – Architecture

- To be Honestly, I think we don't well utilize the ViT model.
- Maybe ViT is too powerful for small dataset.
- Our val-loss can't converge stably and the accuracy is stuck around 0.6 ~ 0.65.

Model Training – Pre-Train dataset

- The ViT is pre-trained on imagenet21k, which is much different from CXR.
- So the feature extractor can't work well on CXR type images.

04.

Conclusion

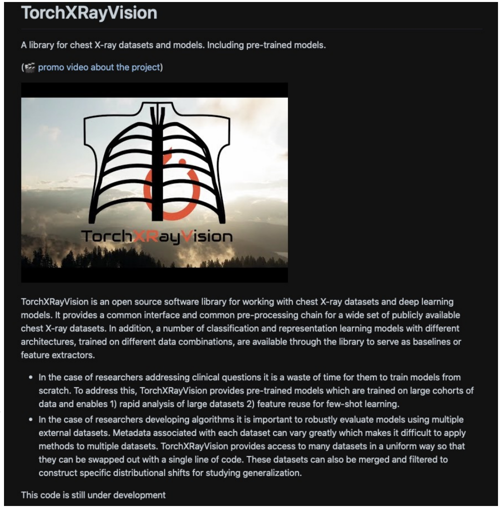
Conclusion

Feature work

- Try to use naïve model like Resnet-50 but pre-train on CXR dataset like (1) Kaggle: RSNA Pneumonia Detection Challenge (2) NIH Chest X-ray dataset (3) CheXpert...
- Use python package : torchxrayvision to train our model.

What have we learning and feedback

- We learn how to deal with CXR images in this project.
- We get some medical knowledge about CXR and CT.
- Use different model and parameter to get better performance.
- It is time-consuming to train such large model and image(528).
- So we will do more survey and well design our experiment.



TorchXRayVision

A library for chest X-ray datasets and models. Including pre-trained models.

(promo video about the project)

TorchXRayVision is an open source software library for working with chest X-ray datasets and deep learning models. It provides a common interface and common pre-processing chain for a wide set of publicly available chest X-ray datasets. In addition, a number of classification and representation learning models with different architectures, trained on different data combinations, are available through the library to serve as baselines or feature extractors.

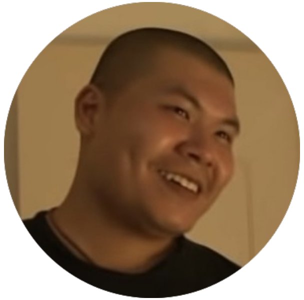
- In the case of researchers addressing clinical questions it is a waste of time for them to train models from scratch. To address this, TorchXRayVision provides pre-trained models which are trained on large cohorts of data and enables 1) rapid analysis of large datasets 2) feature reuse for few-shot learning.
- In the case of researchers developing algorithms it is important to robustly evaluate models using multiple external datasets. Metadata associated with each dataset can vary greatly which makes it difficult to apply methods to multiple datasets. TorchXRayVision provides access to many datasets in a uniform way so that they can be swapped out with a single line of code. These datasets can also be merged and filtered to construct specific distributional shifts for studying generalization.

This code is still under development

Team: CT_06_CCC

OUR TEAM

310551165
Yen-An Chien
簡言安



310551056
Cheng-Ju Ho
何政儒

310551118
Wei-Cheng Chien
簡維成

Team: CT_06_CCC

Member Contribution

Yen-An Chien

25%

- Data Pre-Processing
- Paper/Model Survey
- Beautify the slides
- Github readme writing

Wei-Cheng Chien

25%

- Data Pre-Processing
- Paper/Model Survey
- Beautify the slides
- Github readme writing

Cheng-Ju Ho

50%

- Data Pre-Processing
- Paper/Model Survey
- Model Training
- Experiment Analysis
- Slides & Presentation
- Github

Resources & Reference

Code

- https://github.com/google-research/vision_transformer
- https://medium.com/@kyawsawhtoon/a-tutorial-to-histogram-equalization-497600f270e2
- https://github.com/vicely07/Pneumonet-A-Pytorch-Chest-Xray-Pneumonia-Detection
- https://github.com/hakantekgul/COVID-19_Classification
- https://github.com/mlmed/torchxrayvision

Paper

- COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images https://arxiv.org/pdf/2003.09871v4.pdf
- COVID-19 Image Data Collection https://arxiv.org/pdf/2003.11597v1.pdf
- TorchXRayVision: A library of chest X-ray datasets and models https://arxiv.org/pdf/2111.00595.pdf
- https://www.nature.com/articles/s41598-021-99015-3

Dataset

- https://www.kaggle.com/c/siim-covid19-detection
- https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia
- https://www.kaggle.com/tolgadincer/labeled-chest-xray-images
- https://www.kaggle.com/c/rsna-pneumonia-detection-challenge
- https://www.kaggle.com/c/rsna-pneumonia-detection-challenge

THANKS!

Team CT_06_CCC Members' Email address

CJ Cheng-Ju: ace5271208@gmail.com

YA Yen-An: g0939082708@gmail.com

WC Wei-Cheng: style880810@gmail.com

Our Github Project Link:

https://github.com/luluho1208/NYCU-Digital-Medicine-2021-HW2