COVID X-Ray image classification on modern technique exploration

CSE 6250 student project mideterm draft

Hua Zhao

Abstract

The COVID-19, as a new type of viral pneumonia, has become a pendenmic continuing to threaten the world in 2020. To fight against the desease, first step is to screen it under an easy-carried, fastly-implemented, low-cost low-danger way – typically X-Ray (CXR). More and more researchs that explores and utilizes state of art artifricial intelligence algorithms to experient this job, some of them have been pretty successful in terms of performance. In this research, the author explores some state of art neural networks built by previous researchers, including transfer learning from RestNet[1] and COVID-net[2], under automatic pipelining and big data ETL engineering structure, and tests their perforances. By far (2020-11-16) the first iteration has done; under limited local machine's computing capability, the best trained COVID-Net obtains 0.94 sensitivity for COVID-19 label, given an unlanaced large dataset of 22,049 images collected from various sources, which is by far, according to the best of author's knowledge, the largest dataset used in the COVID-19 CXR classification problem, more than what Linda Wang, et al[2], used in their research with COVID-net. Meanwhile, by feature visualization technique GSInquire, some randome samples are tested, showing features are located within reasonable areas inside lungs. Lastly, the research is still in an ongoing process for future iteration with more computing-efficient techniques to be used, and there are currently problems demonstrated to be sovled along the path.

Keywords: X-ray, COVID-19, classification, algorithm, COVID-net, ResNet

1. Introduction

1.1 Introduce the Problem

There have been some challenges in utilizing neural network algorithms to help with COVID-19 CXR classification. First, it is the limited number of COVID-19 images publically available. Due to the limit of public data, many COVID-19 CXR papers trained their models with less than 500 COVID-19 positive images. The largest collection prior to the paper used in the COVID-19 CXR classification problem is Linda Wang, et al[2] which contains 400+ COVID images with total images of 600+ in 2020-05. The limited datasets cultivate the needs of large and balanced dataset to well-train a complex network to obtain promising classification ability. This makes the reasearch both obtain more authuritative dataset or be more careful with data augumentation when dealing with input dataset. As for obtaining more COVID-19 datasets, Vaya et al.[3] published a new comprehensive dataset including 1,380 CX and 885 DX, which are partly used in this research data collection.

The second issue is CNNs may extract features irrelevant to medical information in CXR images, making the network classifies images with good resutls based on non-medical information from iamges. Maguolo et al[4] in a study trained AlexNet with different sources of COVID CXR datasets where the center images part (mainly the body) were turned to balck, and the network was able to indentify the specific source of dataset with a suprisingly high accuracy. When Tartaglione et al.[5] in their study were training ResNet-18 and ResNet-50, similar things happend when the same source of dataset is used in training and testing - the AUC and accurarcy tend to be very high when training dataset and testing dataset are from same sourc; when training dataset and testing dataset are from different source, those metrics decline significantly - some of the AUCs are even less than 0.50. However, as Bernheim et al.[6] pointed out in a medical study, the main features in COVID CXR images are ground-glass opacity (GGO) and consolidation. This makes training a network with medical-relevant features in CXR essentially important for the purpose of this research.

1) How does the study relate to previous work in the area? If other aspects of this study have been reported previously, how does this report differ from, and build on, the earlier report?

The study is aimed at re-implementing some state-of-art deep neural network models including Linda Wang et el's COVID-Net[2] and Kaiming He et el's ResNet[1, 9] to train on a collected large label-unbalanced dataset. The study differs from previous study in following ways:

1. Data segmentation and historgram quilalization as means to make model focus on searching features

within lung areas in CXR.

2. A large dataset of 22,049 CXR images are used in training, validating, and testing.
3. Spark is used as PySpark package with Python to do ETL for images collection.
4. A DVC pipelining technique manages the dataflow and model reproduction, with a parameter yaml file as input to control all the hypter- and non-hypter parameters used in the production. It utlizies dependency graphs and MD5 checksum to manage data version control, with data serialization in caching.

2. Method

The study utlizies DVC to construct a pipeline that does in order of processing meta data from source images, preprocessing images, loading input hyperparameters, loading model, training, evaluating, outputting performance metrics and trained model with checkpoints.

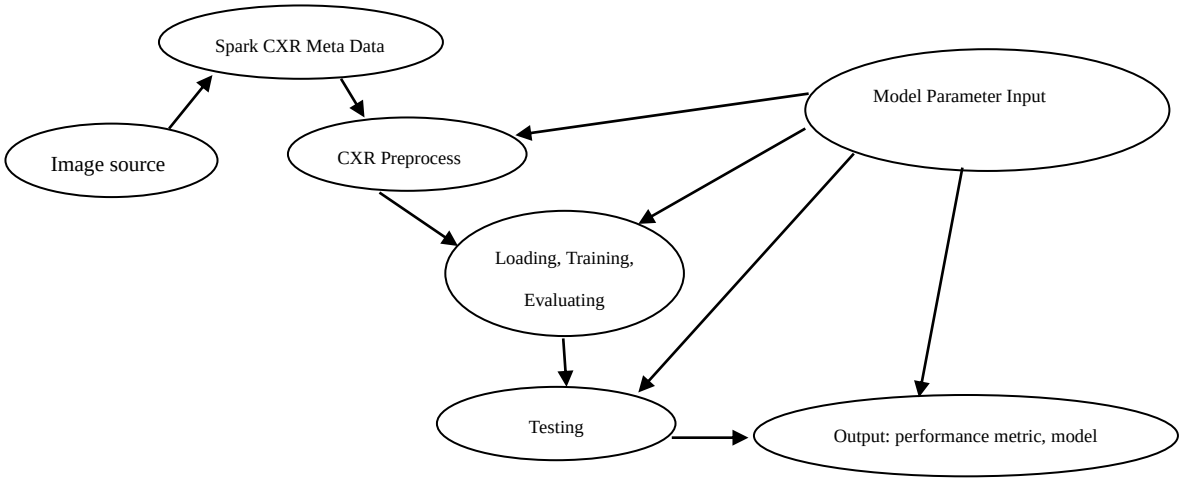
2.0 Source Images

The author used sources that Linda Wang et el[2] suggested. The source was documented in their github[7]. They also provided an ETL process. But the author finds out they didn't fully utilize source images so the author provided a new ETL process with Spark implemented by PySpark and Python that merges source images together. Compared with previous author's 600+ images, the author obtained a dataset with 22,049 images, though COVID images are still small, detailed descriptions displayed as follow:

```
---------------------------------------------------------------------------
                                This research
label distribution:
Counter({'pneumonia': 11092, 'normal': 10340, 'covid': 617})
label integer map:
{'covid': 0, 'normal': 1, 'pneumonia': 2}
test data from covid datasets:
Counter({2: 2773, 1: 2585, 0: 155})
train data from covid datasets:
Counter({2: 8319, 1: 7755, 0: 462})
                                Linda Wang et el[2]
Data distribution from covid datasets:
{'normal': 0, 'pneumonia': 57, 'COVID-19': 617}
test count:  {'normal': 0, 'pneumonia': 5, 'COVID-19': 100}
train count:  {'normal': 0, 'pneumonia': 52, 'COVID-19': 517}
---------------------------------------------------------------------------
```

2.1 Pipeline

DVC uses a dependency graph to reproduce model and manage data version. The graph is shown below:



Specifically in ‘Spark CXR Meta Data’ part, Spark is used to process meta data from over 60,000 source images and cache as a pandas dataframe.

2.2 CXR Preprocess

In additiona to image meta data, during CXR preprocess, images from source files are merged and resized to according to input parameters, currently 224x224x3 is used, but in later phase, the author hopes to use 448x448x3 to see if the resolution comes in play with model performance.

The CXR preprocess also applies historgram qeuilization and data segmentation to help the model catch features within lung areas instead of specific patterns in the image itselt.

Initially the author attemped to process a single large pandas dataframe that restores training and testing image data, cache it for later phase. So in actual training, the program doesn’t have to process images anymore. However, due to limitations of local machine for the program, there are 10% memory leaks when the program loads around 3G image data and does training simultaneously. Therefore, the approach is transferred to process image files into local file system with marked meta data, which saves storage in training.

2.3 Input Parameters

Instead of common CLI most research utilizes to control and mange model inputs, the study uses an external input file in yml extension where you can input which specific model you want to use, pretrained or not, image size, all relevant paths, training epochs, learning rate, display step size, label mapping, decay learning rate parameters, etc. This determines what model the pipeline will use, and how the training will be processed.

2.4 Loading, Traing, Evaluating

2.4.1 Loading

Because pretrained ResNet-50 and COVID-Net-CXR-Small are structured in different program structure, pytorch and tensorflow, the pipeline is able to use both dependent on the specific model. After the pretrained model is loaded with specific checkpoints deired from input file, the model goes to training phase.

2.4.2 Training

During the training, dataset generated from batches are being balanced for COVID-19 class. For tensorflow model, dataset classes is balanced so COVID-19 images are weighted at 0.3 (an input parameter) in each batch. A generator produces batches of image tensors from non-COVID sample pool, at the same time bootstrap from COVID samples and insert it randomly into non-COVID samples to make a label-balanced batch. For pytorch model, dataset classes are perfectly balanced.

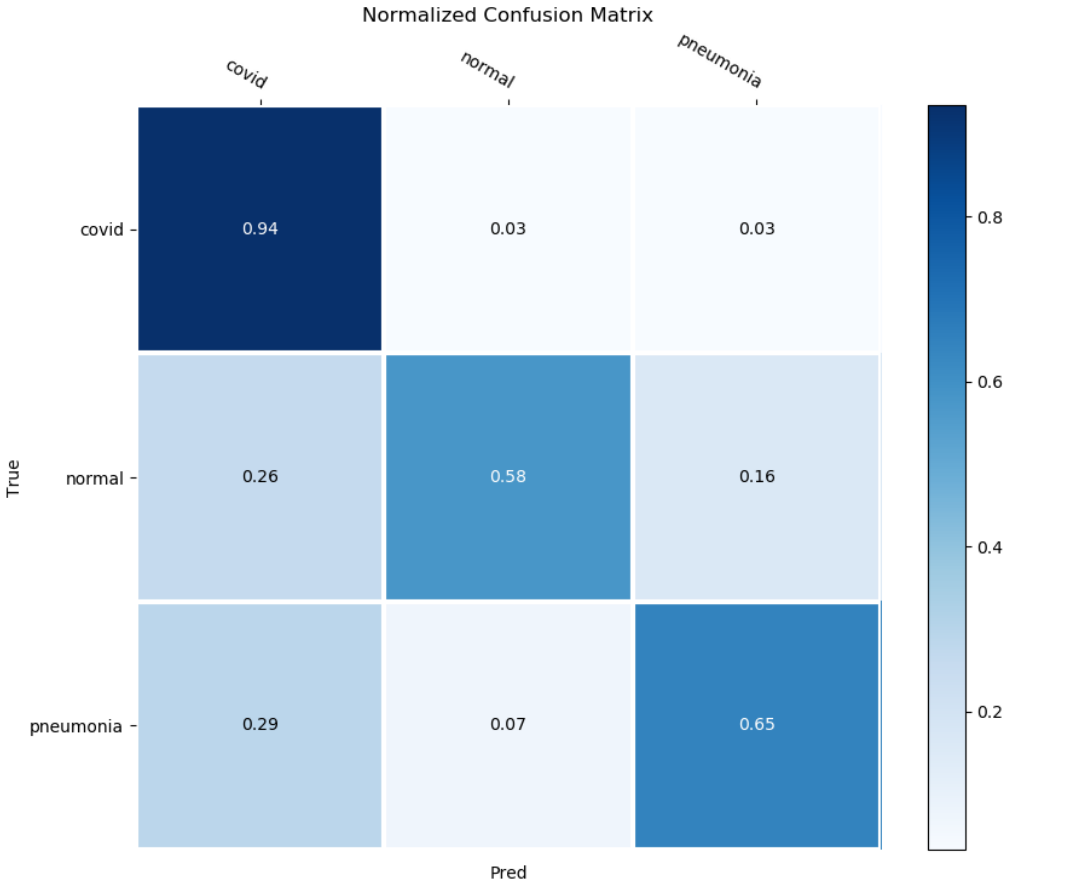
In addition, data augmentation is used in both tensorflow and pytorch model, with horizontal flip rate equals 50%, as well as other random adjustment including rotation, etc. This will make a source image into different tensors in different batches.

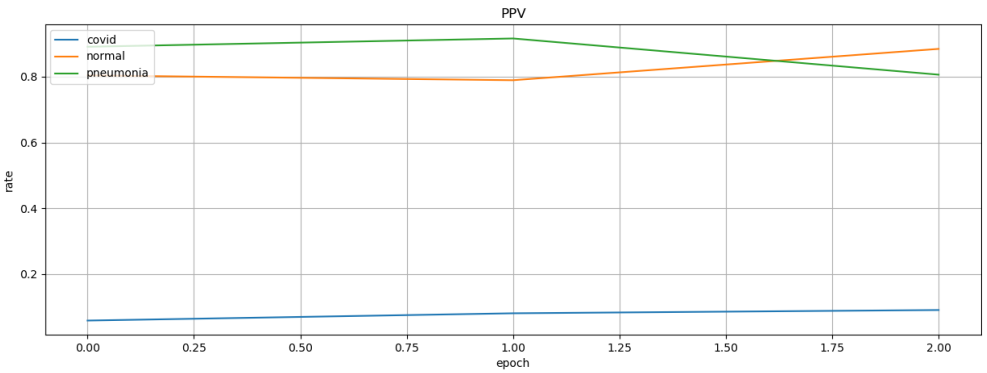
2.4.3 Evaluating, Output

Due to computing limitation under larege image dataset (20,000+), the model is only trained with epochs under 5. Sensitivity, PPV (positive predictive value), normalized confusion matrix are used as performance metrics for the COVID-Net-CXR-Small; losses learning curve, normalized confusion matrix are used as performance metrics for ResNet-50.

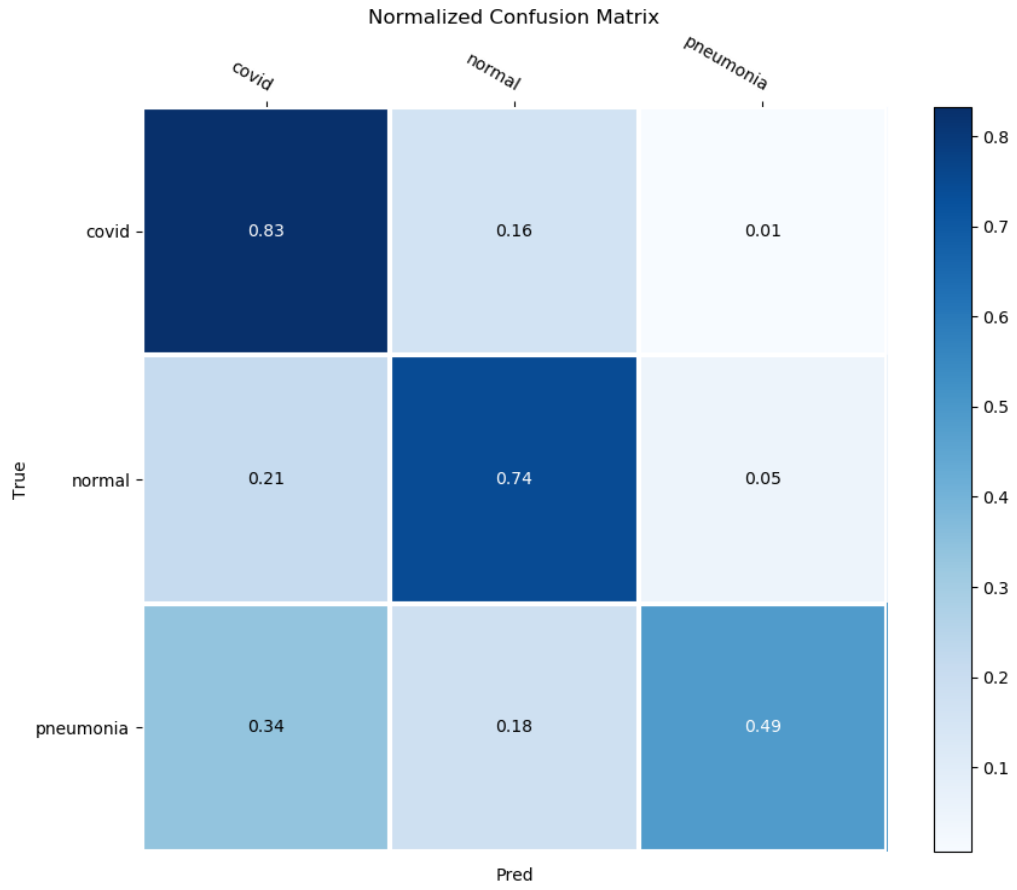
3. Results

For COVID-Net-CXR-Small mode, with 3 epochs, it achieves 0.94 sensitivity score for COVID-19 class, with only less than 0.2 PPV, metrics shown below:





For ResNet-50, as a beseline model, the sensitivity rate is 0.83 for , with relatively higher rates for normal and pneumonia classes.



4. Problems

However, some problems are there to be solved in later phase of the project: mainly exploring and seeking more COVID-19 images from public resources, as well as seeking a more efficient way to train models with more epochs. A new TResNet[8] model will perhaps also be implemented.

References

1. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, arXiv: arXiv:1512.03385
2. LindaWang, AlexanderWong, COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images, arXiv:2003.09871, 2020-05
3. Maria de la Iglesia Vay´a, Jose Manuel Saborit, Joaquim Angel Montell, Antonio Pertusa, Aurelia Bustos, Miguel Cazorla, Joaquin Galant, Xavier Barber, Domingo Orozco-Beltr´an, Francisco Garc´ıa-Garc´ıa, Marisa Caparr´os, Germ´an Gonz´alez, Jose Mar´ıa Salinas, BIMCV COVID-19+: a large annotated dataset of RX and CT images from COVID-19 patients, arXiv:2006.01174v3, 2020-06
4. Gianluca Maguolo, Loris Nanni, A Critic Evaluation of Methods for COVID-19 Automatic Detection from X-Ray Images, arXiv:2004.12823, 2020-09
5. Enzo Tartaglione, Carlo Alberto Barbano, Claudio Berzovini, Marco Calandri, Marco Grangetto, Unveiling COVID-19 from Chest X-ray with deep learning: a hurdles race with small data, arXiv:2004.05405, 2020-04
6. Adam Bernheim, Xueyan Mei, Mingqian Huang, Yang Yang, Zahi A. Fayad, Ning Zhang, Kaiyue Diao, Bin Lin, Xiqi Zhu, Kunwei Li, Shaolin Li, Hong Shan, Adam Jacobi, Michael Chung, Chest CT Findings in Coronavirus Disease-19 (COVID-19): Relationship to Duration of Infection, RSNA Radioalogy, https://doi.org/10.1148/radiol.2020200463, 2020
7. https://github.com/lindawangg/COVID-Net
8. Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, Itamar Friedman, TResNet: High Performance GPU-Dedicated Architecture, arXiv:2003.13630
9. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015