CS598 DL4H Spring 2022 Project Draft

Ashutosh Agarwal and Chandan Goel
aa61@illinois.edu, chandan3@illinois.edu

Group ID: 179
Paper ID: External (Sharma, Rani & Gupta 2020) [7]
Presentation link (TODO): https://www.youtube.com
Code link: https://github.com/AshutoshAgarwal01/CovidPred_Repro

1 Introduction

This work (Sharma, Rani & Gupta 2020) [7] is done to build efficient deep leaning models using the convolutional neural network (CNN) using a very limited set of chest X-ray images to differentiate COVID-19 cases from healthy cases and other types of illnesses.

Several other studies were done before this work to classify COVID-19 patients using chest X-ray images. This study recognizes following issues in classification of COVID-19 based on chest X-ray that result in poor performance of models.

- Scarcity of x-ray images available to train a viable deep learning model.
- Biased learning of the deep learning based model when images of multiple age groups are combined together to form a data set. E.g., x-ray images of pediatric patients combined with adult patients.

In this study, authors have applied following techniques to overcome issues mentioned above:

- To overcome scarcity of x-ray images, authors proposed creating a much larger artificial dataset using smaller original dataset. This artificial dataset was created by applying 25 image augmentation techniques on the original data.
- Careful image selection: Authors propose to use similar type of images (same view, same age group) to train a very targeted model. This study claims that models built with similar type of images perform better than those models that take a wide variety of images.

Overall, authors showed that artificially generated x-ray images using image augmentation techniques greatly improved model performance when compared with original smaller set of images.

2 Scope of reproducibility

"This study shows that a deep learning model trained with larger volume of artificially generated X-ray images using image augmentation techniques will have higher accuracy than the model trained with small set of original images."

Data scarcity is a well-known problem in the field of healthcare analytics, especially when research is aimed at fairly new area e.g., classification of COVID-19 patients when the pandemic just broke out. We are motivated to take on this study as it tries to solve a real life problem which exists across all healthcare domain. It will not only help improve model performance, but also reduce cost and time involved in getting sufficient size data for any healthcare study.

2.1 Addressed claims from the original paper

We will be testing following two claims from the paper.

- **Claim 1**: Model trained with original and augmented images combined results in higher accuracy than model trained with only original images across all classification labels.
  Authors compared performance of following two models on unseen external data.
  - Model with only original images.
  - Model with original images along with augmented images.

  Authors concluded that model trained with original and augmented images combined, results in higher accuracy (True positive rate) across classification labels. In our project work, we want to confirm this claim.
- **Claim 2**: Authors showed that models trained with 120 and 140 degree rotated images were complimentary to each other. i.e., if one model performed bad for certain label than other model would perform better for the same

label. Together, these two models beat performance of model trained with original images. We will verify this claim in our reproduction study.

3 Methodology

We did this work in three phases, namely i) image pre-processing, (ii) data augmentation, (iii) training of deep learning models. The above steps are explained in this section. We leveraged the authors code (GitHub [4]) with moderate modifications for the model training but used our own code for data processing.

3.1 Model descriptions

The author used a simple CNN approach by testing a number of different architectures with different number of layers, number of neurons per layers, normalizations, pooling techniques and hyperparameters and adapting the model after each additional improvement of validation accuracy. The CNN model identified uses 3 convolutional layers followed by two fully connected linear layers.

Figure 1 - manifests the CNN model architecture developed for this paper. Rectified Linear Unit (ReLU) addressed the non- linearity after each convolutional layer.

The first convolutional layer has an input channel of dimension 3 since we are using RGB image of size 256 x 256. The kernel size is chosen to be of size 3x3 with stride of 1. The output dimension at this layer is 32 x 256 x256. This is followed by ReLu activation function to introduce non-linearity. To reduce the number of training parameters and computation cost in conjunction with controlling overfitting issue, a max pooling layer with kernel size 2x2 and stride 2 has been introduced after ReLU activation function. Max pooling layer down sample the output feature maps to 32 x 128 x 128. The second convolutional layer has an input channel of 32 to keep it consistent with the dimension of the feature map from the previous layer. The kernel size is chosen to be 5x5 with stride of 1. The output dimension at this layer is 64 x 128 x 128 so no changes occurred in the transformation channel dimension. Like previous channel ReLU activation function is applied followed by another max pooling layer with kernel size 2*2 with stride 2 is introduced to down sample the feature map led to increased efficiency in model training process. The down-sampled feature map dimension: 64 x 64 x 64.

The third convolutional layer is introduced to upgrade the output channel before feeding into linear layers. Like previous convolutional layers, subsequent maxpool, ReLU activation are applied to the output images. The output dimension at this layer is 128 x 32 x 32. Finally, two fully connected linear layers are used at the end. A flattened version of the feature map is passed to the first fully connected layer. So, the input dimension size has become 128 x 32 x 32 = 131072 nodes. Then this layer is connected to the final linear layer. The output dimension of final layer should match the total image corresponds to five categories: Normal, TB, Pneumonia, Covid-19, Non Covid-19.

3.2 Data descriptions

Data from following three sources is used in this paper.

- Github (Cohen's covid-chestxray-dataset) [1]: This dataset is used to get 'covid 19' and 'non covid 19' images
- Kaggle NIH dataset [2]: This dataset is used to get Pneumonia images.
- National Library of medicine [3]: This dataset is used to get normal and TB images.

Figure 2 depicts all steps we performed to gather and process data for model training and validation purposes.

We filtered images from these datasets using following criteria: i). Age of patient must be 19 years or older, ii) only chest X-ray images and iii) image view must be PA

After filtering the data, we divided the dataset into two parts using random sampling. We reserved 10 percent of the data for external validation (test set) and remaining 90 percent for model training (training set).

After this, we created 25 new datasets by applying different augmentation techniques on original set of training and test images. Some of the augmentation techniques used are:

- Rotate images by 45, 60, 90, 120, 140 and 160 degrees
- Raise blue, green, red and hue
- Crop images
- Flip images horizontally, vertically and in both directions.
- Introduce blur to the images.

We used CloDSA [5] library for image augmentation.

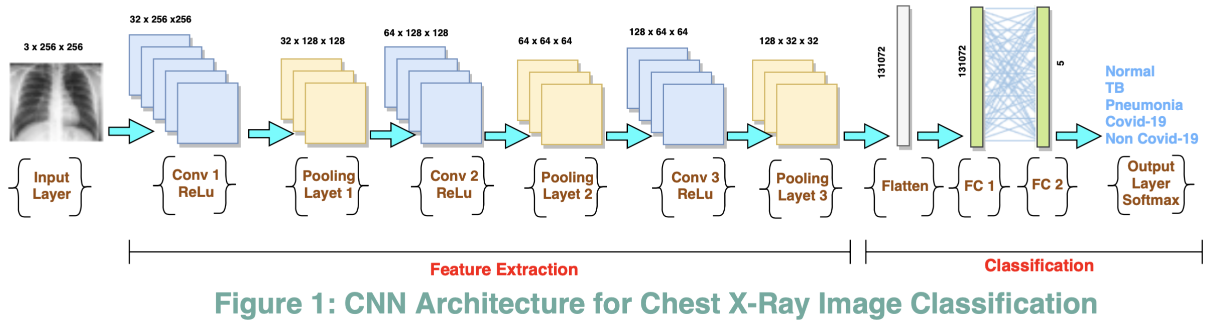


Figure 1: Model architecture

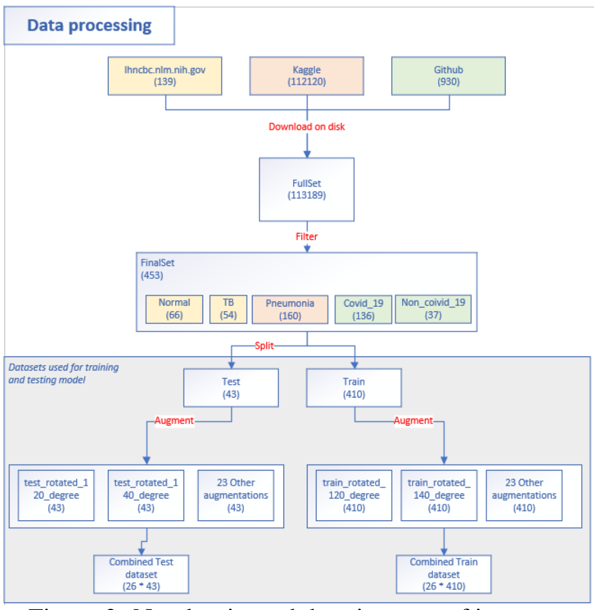

Data processing
lhncbc.nlm.nih.gov (139)
Kaggle (112120)
Github (930)
Download on disk
FullSet (113189)
Filter
FinalSet (453)
Normal (66)
TB (54)
Pneumonia (160)
Covid_19 (136)
Non_coivid_19 (37)
Split
Datasets used for training and testing model
Test (43)
Train (410)
Augment
Augment
test_rotated_120_degree (43)
test_rotated_140_degree (43)
23 Other augmentations (43)
train_rotated_120_degree (410)
train_rotated_140_degree (410)
23 Other augmentations (410)
Combined Test dataset (26 * 43)
Combined Train dataset (26 * 410)

Figure 2: Number in each box is count of images.

We further created one more dataset by combining original dataset and all augmented datasets. Thus we had total 27 datasets.

3.3 Hyperparameters

For initial reproduction work, we used same hyperparameters used in the paper except number of iterations. Table in *Figure 3* describes the hyperparameters and their values across different datasets.

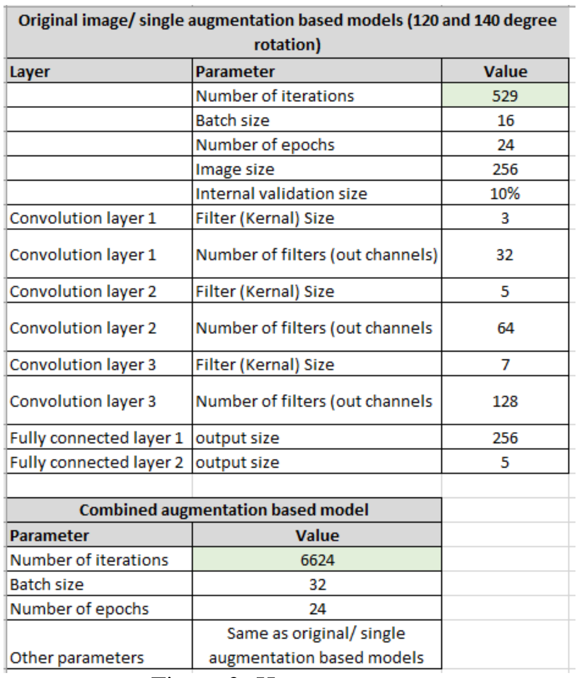
Green boxes represent hyperparameters that were modified for our study.

3.4 Implementation

We thoroughly studied the code available in author's git repository [4]. Following is distribution of code that we wrote and re-used.

- Data processing: We wrote our own code from scratch for data processing.
- Model training and validation: We made some improvements in existing code like logging, refactoring, minor performance improvements and training/ validating models using multiple datasets by using one script etc. But overall, it is heavily inspired by original work.

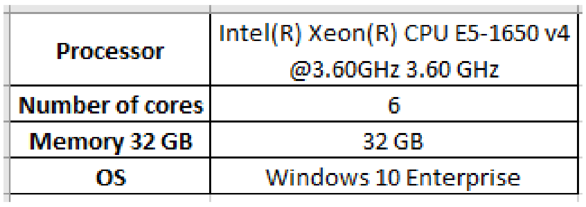
Code of reproduction work can be found in Git repository CovidPred_Repo [6]



Original image/ single augmentation based models (120 and 140 degree rotation)		
Layer	**Parameter**	**Value**
	Number of iterations	🟩 529
	Batch size	16
	Number of epochs	24
	Image size	256
	Internal validation size	10%
Convolution layer 1	Filter (Kernal) Size	3
Convolution layer 1	Number of filters (out channels)	32
Convolution layer 2	Filter (Kernal) Size	5
Convolution layer 2	Number of filters (out channels	64
Convolution layer 3	Filter (Kernal) Size	7
Convolution layer 3	Number of filters (out channels	128
Fully connected layer 1	output size	256
Fully connected layer 2	output size	5

Combined augmentation based model	
Parameter	**Value**
Number of iterations	🟩 6624
Batch size	32
Number of epochs	24
Other parameters	Same as original/ single augmentation based models

Figure 3: Hyperparameters

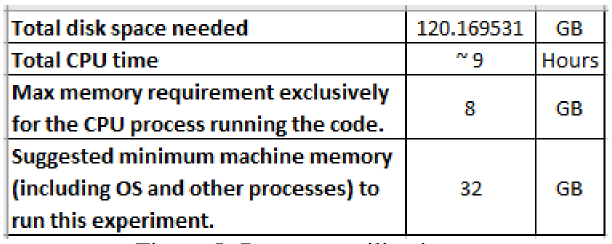


Processor	Intel(R) Xeon(R) CPU E5-1650 v4 @3.60GHz 3.60 GHz
Number of cores	6
Memory 32 GB	32 GB
OS	Windows 10 Enterprise

Figure 4: Configuration of hardware

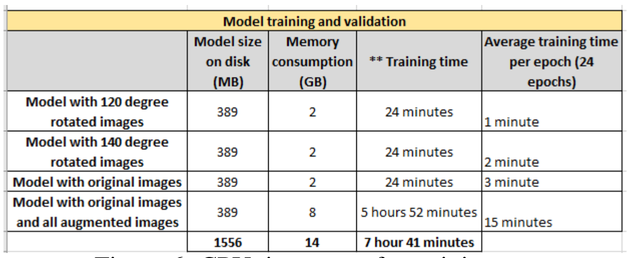
3.5 Computational requirements

In the proposal we mentioned that we will use GPU if training model with large dataset (all augmented images combined) takes very long time. However in our initial runs, even the largest dataset completed in reasonable amount of time. Therefore we decided to use the alternate in-premise CPU based hardware. Tables (*figure 4* and *figure 5*) show configuration of desktop that we used and total resources consumed during work.



Total disk space needed	120.169531	GB
Total CPU time	~ 9	Hours
Max memory requirement exclusively for the CPU process running the code.	8	GB
Suggested minimum machine memory (including OS and other processes) to run this experiment.	32	GB

Figure 5: Resource utilization



Model training and validation				
	Model size on disk (MB)	Memory consumption (GB)	** Training time	Average training time per epoch (24 epochs)
Model with 120 degree rotated images	389	2	24 minutes	1 minute
Model with 140 degree rotated images	389	2	24 minutes	2 minute
Model with original images	389	2	24 minutes	3 minute
Model with original images and all augmented images	389	8	5 hours 52 minutes	15 minutes
	1556	14	7 hour 41 minutes	

Figure 6: CPU time spent for training

Table in *Figure 6* summarizes total CPU time and average CPU time per epoch for each model.

4 Results

Our reproduction study did not result in exact same accuracy numbers as the original study. However overall trend of the accuracy (true positive) per label is consistent with the original study.

4.1 Result 1

The paper claimed that model trained with original and augmented images combined results in higher accuracy (true positive) across all classification labels when compared with model trained with original images only.

Table in *figure 7* summarizes true positive rate (proportions of images correctly classified by the model for given label) of both models on unseen original data.

We can see that model trained with combined dataset outperforms model trained with original images by over 20%. This upholds the paper's conclusion that overall it performs better than the baseline.

However when we look the individual label result, we observed that accuracy for non-covid degrades in combined model. As per our analysis, we feel it happened due to a smaller test dataset size of 3 images. We plan to introduce more non-covid19 images to our model in subsequent study to confirm our analysis.

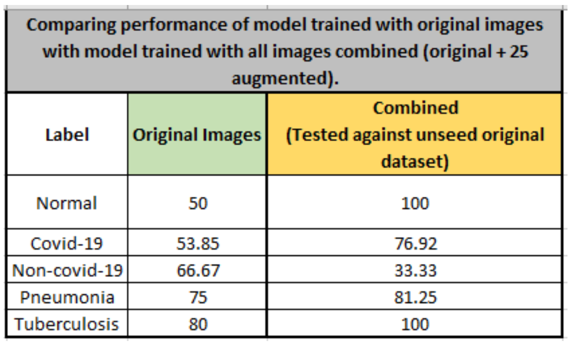
4.2 Result 2

Authors showed that performance of models trained with 120 and 140 degree rotated images were complimentary to each other. Together, these two models beat performance of model trained with original images.

Table in *figure 8* summarizes true positive rate (proportions of images correctly classified by the model for given label) for all three models on unseen original data.

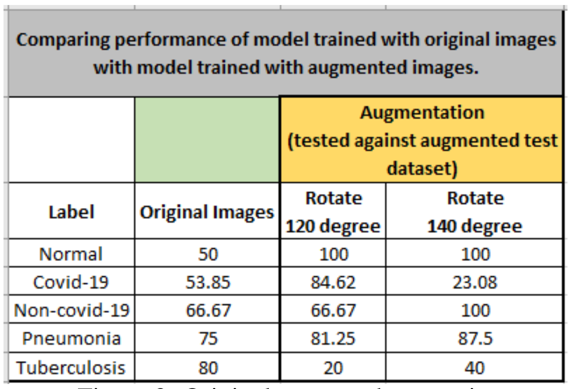
We can see that for all labels except Tuberculosis authors' conclusion holds good. However, both augmented models perform worse than baseline for Tuberculosis. Therefore, this claim is only partially supported by our reproduction study.

We found that the difference is due to random sampling of TB images when compared with original study. We plan to further analyze and test the model with additional random sampling and compare the average result to study the trend.



Comparing performance of model trained with original images with model trained with all images combined (original + 25 augmented).		
Label	Original Images	Combined (Tested against unseed original dataset)
Normal	50	100
Covid-19	53.85	76.92
Non-covid-19	66.67	33.33
Pneumonia	75	81.25
Tuberculosis	80	100

Figure 7: Original vs combined performance



Comparing performance of model trained with original images with model trained with augmented images.			
		Augmentation (tested against augmented test dataset)	
Label	Original Images	Rotate 120 degree	Rotate 140 degree
Normal	50	100	100
Covid-19	53.85	84.62	23.08
Non-covid-19	66.67	66.67	100
Pneumonia	75	81.25	87.5
Tuberculosis	80	20	40

Figure 8: Original vs rotated comparison

4.3 Additional results not present in the original paper: TODO

5 Discussion: TODO

5.1 What was easy: TODO

5.2 What was difficult: TODO

5.3 Recommendations for reproducibility: TODO

6 Communication with original authors: TODO

References

1. Cohen's covid-chestxray-dataset: https://github.com/ieee8023/covid-chestxray-dataset
2. Kaggle NIH dataset: https://www.kaggle.com/datasets/nih-chest-xrays/data
3. National Library of medicine: https://lhncbc.nlm.nih.gov/LHC-publications/pubs/TuberculosisChestXrayImageDataSets.html
4. CovidPred: https://github.com/arunsharma8osdd/covidpred
5. CloDSA: https://github.com/joheras/CLoDSA
6. Reproduction work: https://github.com/AshutoshAgarwal01/CovidPred_Repro
7. Original Paper: https://www.hindawi.com/journals/ijbi/2020/8889023/