

**Deep Learning**

***Covid Patients Chest X-Ray Image Interpretation***

***Team 2***

*Fady Smouni*

*Nidhi Danayak*

**School of Graduate Professional Studies**

Data Analytics

DAAN 570 – Deep Learning

Spring II, 2021

**Document Control**

**Work carried out by:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Email Address** | **Exhaustive list of Tasks** |
| Fady Smouni | ffs5130@psu.edu | Models, data extraction, paper, presentation |
| Nidhi Danayak | nmd5606@psu.edu | Models, data extraction, paper, presentation |
|  |  |  |

**TABLE OF CONTENTS**

[1 Introduction 3](#_Toc33129402)

[2 Problem Statement 3](#_Toc33129403)

[2.1 Challenges 3](#_Toc33129404)

[2.2 Related Works 3](#_Toc33129405)

[2.3 Importance 3](#_Toc33129406)

[3 Data Collection 3](#_Toc33129407)

[4 Data Preprocessing 3](#_Toc33129408)

[5 Methodology 4](#_Toc33129409)

[6 Results 4](#_Toc33129410)

[7 Discussion of Results 4](#_Toc33129411)

[8 References 5](#_Toc33129412)

# Introduction

The Coronavirus Disease 2019 (COVID-19) pandemic continues to have a devastating effect on the health and well-being of the global population. A critical step in the fight against COVID-19 is effective screening of infected patients. It was found in early studies that patients present abnormalities in chest radiography images that are characteristic of those infected with COVID-19. Motivated by this and inspired by the open-source efforts of the research community, in this study we try to build a deep Convolutional Neural Network to detect COVID-19 cases from chest X-ray (CXR) images that is open source and available to the general public. This open source effort, represent a great opportunity for many individuals to come together and work on projects that are for the greater good.

# Problem Statement

While RT-PCR testing as the standard test for COVID-19 detection, it is a very time-consuming, laborious, and complicated manual process. Besides studies show that there are high chances of getting a False Positive with RT-PCR. To make this detection faster, Group 2 tries to build a Deep Convolutional Neural Network which will predict the presence of Coronavirus using the patient’s chest x-ray image, so that only patients with a Positive result can be taken further for an RT-PCR test.

# 3 CHALLENGES

* Some of the main challenges we faced came with accessibility to data. Medical data is very hard to get. This is due to laws governing patient privacy. Many of the datasets we found were too small to use for any type of meaningful training of our neural network.
* Another major challenge we faced was the limitations that come with our computational resources. This represented a serious hurdle for us while building our models, as we needed long hours to train them. However, at the end, we were able to use GPUs instead of CPUs and that allowed us to speed up the process quite a bit.
* Lack of labeled data. It would be hard for non-medical professionals to label medical imaging datasets. Apart from image classification to detect COVID-19 patients, we also wanted to perform Image Segmentation which would create a boundary box around the area of infection. Unfortunately, there was a lack of such dataset and would have required a medical professional’s assistance to manually segment the images accurately.

## RELATED WORKS

* The dataset was extracted from [iCTCF](http://ictcf.biocuckoo.cn/HUST-19.php) website by Huazhong University of Science and Technology.
* This research group had used this dataset along with each patient’s 130 clinical features to build a novel framework of Hybrid-learning for UnbiaSed predicTion of COVID-19 patients (HUST-19) to predict the outcomes along with the Morbidity and Mortality outcomes.
* To our knowledge, the other research studies in this field try also try to classify chest x-ray images to identify Covid-19 patients. But with the field of Deep Neural Network being so vast, there can never be a single best model available. We therefore try to train the dataset in multiple State of the Art CNNs to get the maximum achievable accuracy.

## IMPORTANCE AND IMPACTS

* The deadly COVID-19 virus has rapidly spread into multiple countries worldwide. With the mortality rates going high every day, early diagnosis becomes a key element for proper treatment of the patients and prevention of the spread of the disease.
* By leveraging the identification of this virus’s lung involvement in infected patients we try to build this COVID-19 detection model. This model will eliminate the need of manually checking the X-Ray scans to detect virus’s presence. And thus, will help the medical facilities quickly detect a positive coronavirus patient and provide them the necessary treatments which could save more lives.
* This project will also allow for the creation of even more common work between medical professionals and the deep learning community. Allowing for greater access to data in the future and showing the medical community, the importance of deep learning and the major benefits it can bring to the table.

# Data Collection

* Source(url): http://ictcf.biocuckoo.cn/HUST-19.php
* The data set of iCTCF is an open resource dataset that contains chest computed tomography (CT) from 1170 patients. We are provided with 19,685 manually labeled CT slices. The labeling splits them into 3 categories; non-informative CT (NiCT), positive CT (pCT), negative CT (nCT).
* The 19,685 labeled CT slices consist of 5705 NiCT, 4001 pCT, and 9979 nCT.
* NiCT slices are scans where the lung parenchyma was not captured therefore, not allowing the medical professionals to discern Covid-19 Pneumonia.
* pCT slices are scans where imaging features associated with COVID-19 pneumonia could be clearly discerned.
* nCT slices are scans where imaging features were not associated with COVID-19 pneumonia.
* This dataset plays a major role in our goal to predict COVID-19 pneumonia using CT scans. We are able to use this labeled data to train our model to discern between CTs that show positive COVID-19 pneumonia features and also negatives ones. Moreover, this dataset allows us to also train our model to detect CTs that it should not predict COVID-19 pneumonia on when there isn't enough information on the CT scan, or it is an incomplete CT scan. This is an important addition to our model, as it is important to know when not to make predictions, especially in the medical field when the stakes are very high.

# Data Preprocessing

* The first step while building a model using image as inputs is to load the jpeg images and convert them into pixels because a model only understands image pixels.
* This step consists of getting the path to our data and shuffling the paths in order to avoid our model becoming biased to the order the images are loaded in. We set a seed of 97 for the shuffle to be reproduced by others.
* We used cv2 library to resize the images to 64x64 while loading in our memory. This would not overwhelm our memory and GPU since we have 19,685 images.
* We then extract the label for each image from its path.
* We also scale the pixels to become between 0 and 1 and create numpy array of both the images data and the labels.
* Another important part of our data preprocessing consists of data augmentation. This allows us to improve the performance of our networks by providing them with artificially created new training data. For image data augmentation, this involves the random flips, zooms, rotations to try and cover more of the possible variations that images can be seen in.

# Methodology

* In this project, we are going to use convolutional neural networks (CNN or ConvNet). This deep learning architecture could be labeled as one of the most popular ones out there. Even though this architecture style has been around since the late 90s when introduced by Yann LeCun, it only grew in popularity around 2012 with the introduction of AlexNet. Since then, we have seen the emergence of many models that build deeper and more complex CNN architectures while increasing the accuracy outputs, especially during the ImageNet Large Scale Visual Recognition Challenge which gave birth to many state of the art models that are used nowadays in image classification and that we will use in our own project. CNN models are very powerful as they are able to learn distinguished features of each label (category) of an image.
* In our case, we opted to start by setting up a simple CNN to set up a baseline to improve upon. As our goal is to match the state of the art result that Wanshan Ning[1] was able to achieve 99% accuracy in his co-authored paper about prediction of COVID-19 pneumonia via deep learning.
* General models guideline
  + All our ConvNets will be configured to take an input shape of (64, 64, 3). We format our images this way, because of the limitations of our resources. In the future, with access to more powerful computational resources, we can increase the resolution of our images. But for now, any major increase in the resolution of our images would result in out of memory problems because of the size of our dataset and how deep ad rich in parameters some of our networks are.
  + All of our ConvNet models will be flattened to be fed into a densely-connected network that has a few dense layers and a final activation layer "softmax" for a multiclass classification (NiCT, nCT, pCT)
* Model 1
  + We first decided to set up a simple CNN model that will serve as a baseline. It consists of 6 layers of stacked Conv2D and MaxPooling2D. We then flatten it to set up our fully connect layers.
  + This model is very small in comparison to the state of the art models that are out there. We only have 1,053,795 trainable parameters.
  + We can see below the limitations of this model during its evaluation.
* Model 2
  + The second model that we build is inspired from the VGG19 architecture. This architecture is popular because of its high performance on the imagenet dataset competition. However, it does come with its limitation. It is heavier and requires longer training time as it has a large number of parameters to train.
  + We are able to get high accuracy performance with this model. However, we believe the training time to accuracy tradeoff is not worth it.
* Model 3
  + The third model we use relies on transfer training. We import the DenseNet121 model using keras. The DenseNet121 architecture has been shown to perform well on chest X-rays in Rajpurkar co-authored paper Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. This model allows us to save on time and get comparable accuracy to the VGG19 model we ran previously.
  + When we try and add a few dense layers into this model, it ends up underperforming in comparison to just running it with its original architecture. It is very hard to modify the structure of a model to get better results when the model already performs at 99% accuracy without any modifications.
* Unshown models
  + We also ran a few transfer training models using ResNet50, InceptionV3, VGG16 for testing purposes and to compare their performance to our other models. These models are not shown here to not increase the time it would take to run this jupyter notebook. The difference in performance with our used models was marginal, except for VGG16 which underperformed heavily.
* Future Models
  + We would like to take this project further by combining clinical data about the patients with their computed tomography to predict severity of COVID-19 infection (mild, normal, severe, critical). This will allow us to go beyond just diagnosing an individual as COVID-19 positive or negative. It would allow us to create a system that can help prioritize and redirect medical resources based on the severity of cases and their distribution around the region.
* To train, validate, and test our models, we split our data into a 80:20 split for training and testing and after that split the training data into a 90:10 split for training and validation.
* We use the Adam optimizer as it is a combination of both AdaGrad and RMSProp. It is able to compute individual adaptive rates for different parameters from estimates of first and second moments of the gradients. It has been widely accepted in the CNN community for its great performance and results.
* Convolutional Neural Networks also represent a major step forward compared to traditional machine learning techniques. They do not need a person to manually perform feature extraction and tell it what is important and what is not. CNNs are able to find the important features on their own. This is a huge advantage as it gets rid of the human factor that can be either expensive to hire or mistake prone.

# Results and INterpretation

* Model1:
  + As we can see, model1 performs the least well, with an 87% accuracy. This model is a simple CNN model that is used to set a baseline of what is possible and improve upon.
  + It is relatively very fast to train, only 12 minutes as shown in table 1.1.
  + As we can see from the image 1.1, this model overfits heavily on the training data, but has a hard time with the validation dataset. That is confirmed even more when we run the test data prediction and get an accuracy of 87% as shown in table 1.2.
* Model2:
  + As we can see, model2 performs much better than the first model. We are able to achieve a 99% accuracy. This is no surprise, as this type of VGG19 architecture has been proven to work very well on image classification over the past years. Getting to achieve a 99% accuracy, it is hard to make any more changes to the model to tweak it in order to get better accuracy, as that is not possible.
  + However, a big downfall of this model as stated above is that it have many parameters (35,768,387) and takes a long time to train. As can be seen in table 1.1, it took 64 minutes to train this model.
  + However, this long training period seems to be worth it as we are able to get great accuracy as show in table 1.3 and also very little to no overfitting as can be seen in image 1.2.
* Model3:
  + As we can see, model3 performs much better than the first model and as well as the second model. We are able to achieve a 99% accuracy. This is no surprise, as this type of DenseNet121 architecture has been proven to work very well on image classification over the past years. Getting to achieve a 99% accuracy, it is hard to make any more changes to the model to tweak it in order to get better accuracy, as that is not possible. Moreover, since we imported this model, when we tried to add a few dense layers to this model, we were left with lower accuracy than just using the model architecture as is.
  + This model also makes up for the long training time of model2, it only has 6,953,856 to train. As can be seen in table 1.1, it took 32 minutes to train this model.
  + However, this shorter training period seems to not be without its downfall, we can see from image 1.3 that we have some slight overfitting compared to model2 as we see that out training and validation accuracy do not converge as smoothly.
* So, when it comes to choosing between model2 and model3, it really depends on the goal of the user. If some overfitting is acceptable for faster results, then that is definitely a small concession to make to double the speed of training. One needs to keep in mind that as we add more images to our training set, the training time will grow longer and longer. We also need to keep in mind that the second models have about 5 times the number of parameters to train when compared to model3. This also needs to be taken into consideration as it means we will a lot more memory to train it. So, check your computational abilities and memory size to make sure that you can handle the bigger model.

|  |  |
| --- | --- |
| **Model** | **Total Training time** |
| Simple CNN | 12 |
| VGG19 | 64 |
| DenseNet121 | 32 |

|  |
| --- |
| Tables 1.1 |

|  |
| --- |
| Tables 1.2 |

|  |  |  |
| --- | --- | --- |
| **Simple CNN** | **Precision** | **Recall** |
| NiCT | 1 | 0.71 |
| nCT | 0.89 | 0.97 |
| pCT | 0.74 | 0.98 |

|  |
| --- |
| Tables 1.3 |

|  |  |  |
| --- | --- | --- |
| **VGG19 architecture model** | **Precision** | **Recall** |
| NiCT | 1 | 0.98 |
| nCT | 0.99 | 0.99 |
| pCT | 0.97 | 0.99 |

|  |
| --- |
| Tables 1.4 |

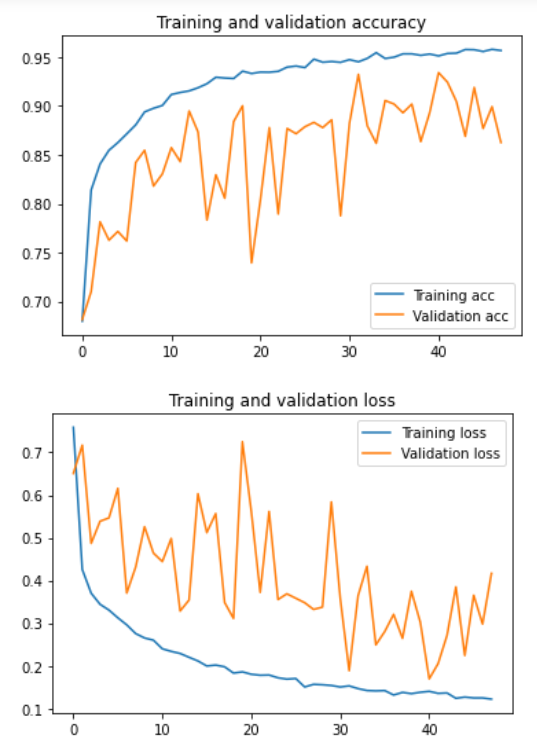
|  |  |  |
| --- | --- | --- |
| **DenseNet121 transfer training model** | **Precision** | **Recall** |
| NiCT | 1.00 | 0.99 |
| nCT | 0.98 | 0.99 |
| pCT | 0.97 | 0.98 |

|  |
| --- |
| Tables 1.5 |

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| Simple CNN | 0.87 |
| VGG19 architecture model | 0.99 |
| DenseNet121 transfer training model | 0.99 |

|  |
| --- |
| Image 1.1 |

|  |
| --- |
| Image 1.2 |

Graphical user interface

Description automatically generated

|  |
| --- |
| Image 1.3 |

Graphical user interface, chart

Description automatically generated

# Discussion of Results

**DISCUSSION OF RESULTS**

* Many state of the art CNN architectures are available to tune and fit to different image classification problems.
* While a simple Convolutional Neural Network can give a commendable accuracy (87%), the architecture heavy state of the art networks are always a better option because they have already been pretrained on huge, varying datasets.
* This project also helped us realize the Importance of large datasets to train a model
* With this project we hope to provide the medical community, an automated system for coronavirus detection. This automated process will not only reduce radiologist’s manual labor on COVID-19 detection using chest X-Rays, but will also speed up the detection time.

**LIMITATIONS**

* The major limitation that came with our project was the initial unavailability of a heavy processing unit. This was later fixed my incorporating our Jupyter notebooks with GPU.
* One of our aims out of this project was to also prepare a model for segmenting the infected regions of lungs. The lack of availability of such labelled data limited our approach to just image classification

**FUTURE WORK**

* For future we aim to combine the imaging prediction model with patient’s clinical data to check if these features are correlated with the severity of COVID-19 symptoms
* We also aim in finding segmented chest X-Ray data to build an image segmentation model along with the current image classification model.

# Your Feedback

* The high computational power of Deep Neural Network can only be leveraged through GPU. Thus, in our opinion availability of a GPU is the first requirement for training such huge datasets using as Deep Neural Networks as the State of the Art works.
* There is never a perfect model when it comes to data mining. Our results and analysis are therefore open to future enhancements.

# References

[1] Wang et al. ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. <https://nihcc.app.box.com/v/ChestXray-NIHCC/file/256057377774>

[2] Dataset was extracted from : <http://ictcf.biocuckoo.cn/HUST-19.php>

[3] Cohen et al. Chester: A Web Delivered Locally Computed Chest X-ray Disease Prediction System. 2020. Proceedings of Machine Learning Research – Under Review:1–12, 2020

[4] Wang et al. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X‑ray images. (2020) 10:19549. https://doi.org/10.1038/s41598-020-76550-z

[5] Tuan D. Pham. Classifcation of COVID-19 chest X-rays with deep learning: new models or fine tuning? 2021. https://doi.org/10.1007/s13755-020-00135-3

[6] Tartaglione et al. Unveiling COVID-19 from CHEST X-Ray with Deep Learning: A Hurdles Race with Small Data. Int. J. Environ. Res. Public Health 2020, 17, 6933. doi:10.3390/ijerph17186933

[7] Wanshan et al. Open resource of clinical data from patients with pneumonia for the prediction of COVID-19 outcomes via deep learning. 2020. Nature Biomedical Engineering | VOL 4 | December 2020 | 1197–1207 | [www.nature.com/natbiomedeng](http://www.nature.com/natbiomedeng)