PROJECT PROGRESS REPORT FOR CLASSIFICATION OF LUNG DISORDERS

Dhrumil Patel
Student# 1007185181
da.patel@mail.utoronto.ca

Kai Richard Topaloglu
Student# 1001553687
hakan.topaloglu@mail.utoronto.ca

Kevin Ge
Student# 1004818361
kevin.ge@mail.utoronto.ca

Jacky Yang
Student# 1004803616
yacky.yang@mail.utoronto.ca

ABSTRACT

This project aims to develop a deep learning tool that can quickly and accurately classify X-ray images of lungs as normal, pneumonia, or COVID-19 cases. With the COVID-19 pandemic, the need for accurate and efficient diagnosis is pressing, and traditional methods can be time-consuming. We are using convolutional neural networks (CNNs) to develop the model, and the input is an X-ray image with a classification label as the output. We have already achieved significant progress in creating a baseline model with accuracy of 92.8 percent and a rudimentary primary model with 94 percent.

—-Total Pages: 8

1 PROJECT DESCRIPTION

The motivation behind our project is to develop a tool that can accurately classify X-ray images of lungs as normal, pneumonia, or COVID-19 cases. With the current COVID-19 pandemic, there is a pressing need for accurate and efficient diagnosis, which can aid in timely treatment and control of the disease. Traditional methods of COVID-19 diagnosis, such as polymerase chain reaction (PCR) testing, can be time-consuming and have limited availability. On the other hand, chest X-ray can provide quick results, making it a promising tool for COVID-19 screening and diagnosis. Furthermore, COVID-19 and pneumonia have very similar symptoms, but a chest X-ray can tell the difference. Therefore, the ability to distinguish between COVID-19, pneumonia, and normal lungs can help reduce the likelihood of misdiagnosis, which can lead to inappropriate treatment and worsen a patient's health.

The goal of our project is to develop a deep learning model that can accurately classify X-ray images of lungs into three categories: normal, pneumonia, and COVID-19. By training a deep learning model on a large dataset of labeled X-ray images, the model can learn to distinguish between the different categories and classify new images with high accuracy.

Deep learning is a reasonable approach for this task because it has shown impressive results in image recognition and classification tasks. Convolutional neural networks (CNNs),have been used for various medical imaging analysis tasks as they can extract relevant features from the input images, automatically learn new features and accurately classify input images. This can improve the accuracy of classification. Additionally, CNNs are efficient at processing large amounts of image data, making them a practical approach for this problem.

The input of our model will be an X-ray image of a lung (augmented from original data set), and the output will be a classification label indicating whether the image is of a normal lung, a lung with pneumonia, or a lung with COVID-19. Our model will be trained on a large dataset of labeled X-ray images of lungs and validated on a separate test set to ensure its accuracy and generalizability. A visual overview of our project can be seen below in Figure 1.

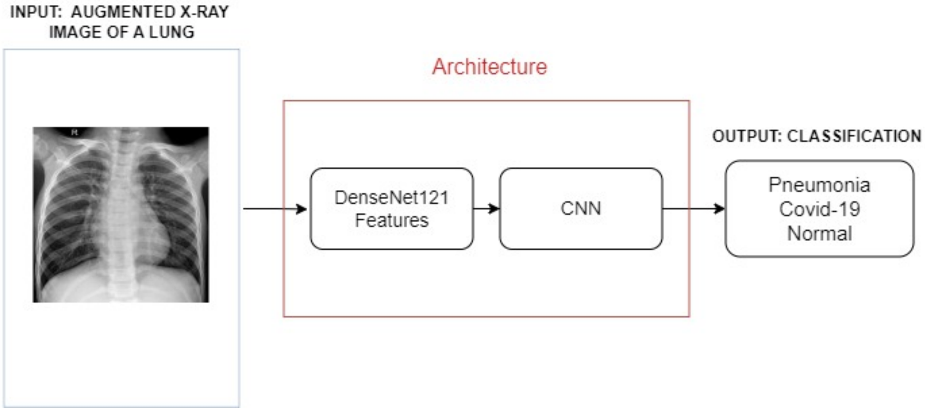


Figure 1: Model Overview

2 Individual Contributions and Responsibility

Our team is working together in multiple ways to ensure we stay on track to finish the project before the deadline and meet each deliverable. We have met every Tuesday to discuss about the project, divide tasks and update each other on our progress. This has also been a great opportunity to bond as a team and learn from each other. Furthermore, we have utilized GitHub, Git and Google Collab to track and share our code. Our main mode of communication to share insights and problems has been through Discord. Discord has been a great platform for real-time communication, virtual meetings and ease of use. So far, our team has created an augmented set of training and validation data. We have also created a baseline model and rudimentary primary model with a relatively high accuracy, which we will improve in the next couple of weeks by tuning the hyper parameters. Table 1 shows specifc tasks done by each team member and internal deadlines for future tasks.

Table 1: Tasks Updated (March 17)

Name	Task	Internal Deadline
Dhrumil	Project Plan and Risk Register Part for Proposal	Done
	Contributions/Introduction part of the progress report	Done
	Extract iamges from data source, split, resize, ready to be inputted into the model	Done
	Train data and fine tune parameters	March 24th
	Quantitative/Qualitative Results for presentation	April 5th
	Quantitative/Qualitative Results for final report	April 14th
Kai	Intro and Background of Proposal	Done
	Augment images to be inputted into the model	Done
	Data Processing Part of Progress Report	Done
	Tune Hyper parameters	March 24th
	Intro/Conclusion/Problem/Model parts of presentation	April 5th
	Intro/Conclusion/Problem/Architecture/Ethical parts of final report	April 14th
Kevin	Data Processing and Illustration Section of Proposal	Done
	Develop Baseline model	Done
	Baseline part of the progress report	Done
	Data Processing part of the presentation	April 5th
	Data Processing part of the final report	April 14th
Jacky	Architecture, baseline and Ethical Section of Proposal	Done
	Develop Elementary Primary Model for Progress Report	Done
	Primary Model part of progress report	Done
	Evaluation of results and discussion part for presentation	April 5th
	Evaluation of results and baseline model part of final report	April 14th

3 Data Processing

Based on our research, the majority of projects in this field leverage a combination of chest X ray images and CT scans of the affected patients. Although these types of data sets could have been used to develop our model, we believed the pro of obtaining a more predictious model did not outweigh the con of false reporting. It's a well known issue in the medical imaging industry that CT scans may also produce more false positives or false negatives than chest x-rays, depending on the specific imaging protocols and interpretation methods used. This can introduce additional challenges for accurate diagnosis and classification. As a result, we decided to only aggregate chest X ray images in our dataset.

3.1 Data Sources

Since our model is a CNN, we believed using images would be the best method to train it. Data was aggregated and cleaned from the NIH chest X ray dataset and the following Kaggle data set.

3.2 Data Cleaning and Augmentations

Since the data of interest was split across 2 external data sources, our first goal was to aggregate appropriate data into 2 distinct datasets(1 used for training/validation and the other used for testing). With respect to the NIH chest X ray dataset, chest X rays of interest were conveniently under a "pneumonia", "covid", "control" column. We then sliced theses columns for form our testing dataset. Our training/validation dataset has a size of **5155** X rays.

At a surface level, these were the data augmentation steps we implemented in order:

1. Removed irrelevant images, such as those that were not chest X-rays or those that were duplicates.
2. Split the primary kaggle dataset into training and validation sets with a 80/20 split.
3. Resized all images to 224x224, applied normalization using the mean and standard deviation values computed for each classes, applied random image crops without replacement since our overall dataset size is low.
4. Balanced the dataset using a weighted sampler

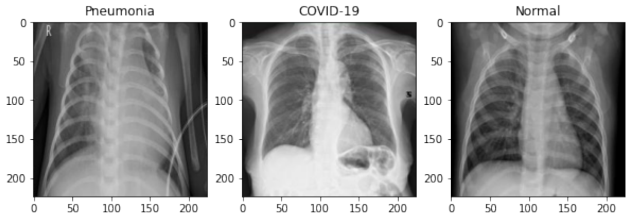


Figure 2: Image sample for each class post data augmentation

Expanding on how we balanced the dataset , we noticed that there was a statistically significant distribution difference with respect to each of our classes.

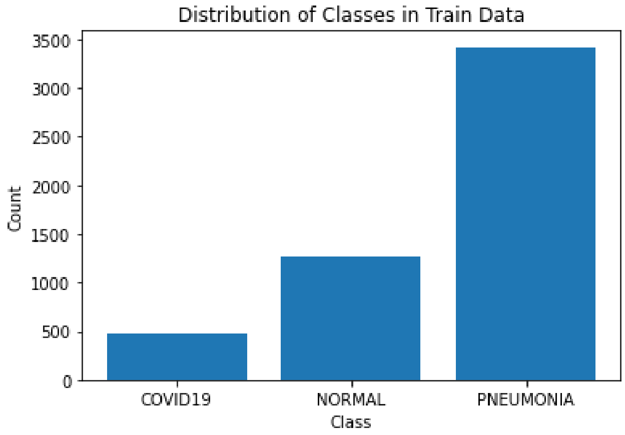


Figure 3: Training set class sample counts before weighted sampler

In order to fix this issue, we applied random weighted oversampling on the COVID19 and NORMAL classes and under sampled the Pneumonia class. This left us with an approximately equal distribution (**1703, 1739, 1713** for COVID19, NORMAL and PNEUMONIA, respectively).

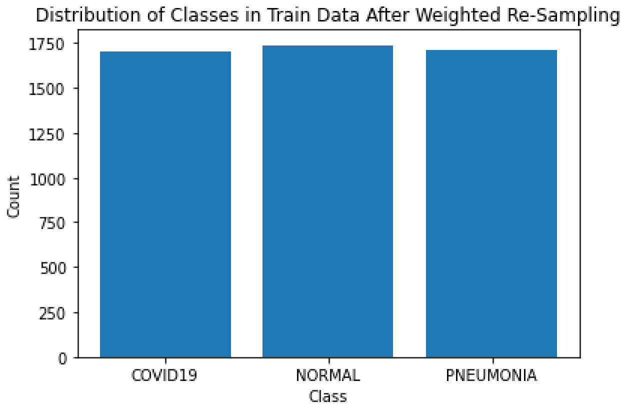


Figure 4: Training set class sample counts after weighted sampler

3.3 Testing Plan

As mentioned previously, the testing dataset was sampled from a different source than the training/validation. The split was 80/20 for the training, validation respectively. Now referencing the cleaned NIH chest X ray dataset, we sampled an equivalent amount of images with respect to our validation data set.

The NIH chest X ray dataset was chosen to represent the data source of our testing set because it is representative of real-world scenarios, is diverse and covers a range of conditions that the model is expected to encounter. This includes variations in image quality, lighting conditions, patient demographics, and disease severity. For example, the original NIH chest X ray dataset contains **112,120** X-ray images with disease labels from **30,805** unique patients, making it very suitable for testing.

3.4 Challenges

Class imbalance: The original datasets were highly imbalanced, with a significant number of samples for the pneumonia class and relatively few samples for the normal and covid19 classes. To address this challenge, we used a weighted sampler to oversample the minority classes while undersampling the majority class. This approach helped to balance the class distribution, thereby reducing the risk of the model being biased towards the majority class.

Selecting appropriate data augmentations: The objective here was to increase the diversity of the training data, which can help to improve the model's accuracy and generalization. One challenge we faced with data augmentation was determining the appropriate transformations to apply. For example, would it make sense to apply random rotations since in the real world, a doctor would never diagnose a patient's chest X ray that isn't upright.

4 Baseline Model

We used a SVM (Support Vector Machine) model that is trained on a subset of our cleaned xrays to classify the xrays images into 3 classes: COVID19 Pneumonia, Pneumonia or Normal. We use an RBF (radial basis function) kernel and the decision function shape is one-vs-rest (ovr) to handle non-linearily separable data and multi-class classification, respectively.

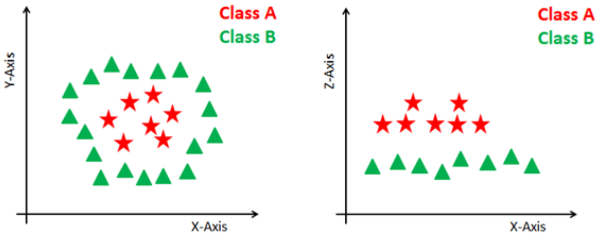
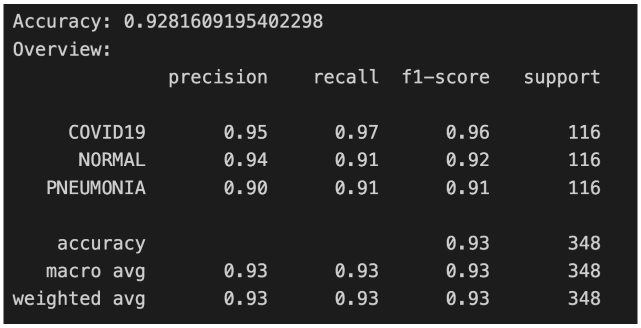


Figure 5: SVM transforms the (non-linearly separable) input space to a higher dimensional space as shown on the right

Before we analyze the results, here are the quantitative results we got from sckit-learns metrics class:



```
Accuracy: 0.9281609195402298
Overview:
              precision    recall  f1-score   support

     COVID19       0.95      0.97      0.96       116
      NORMAL       0.94      0.91      0.92       116
   PNEUMONIA       0.90      0.91      0.91       116

    accuracy                           0.93       348
   macro avg       0.93      0.93      0.93       348
weighted avg       0.93      0.93      0.93       348
```

Figure 6: Our SVM results

The results obtained from the model show an accuracy of 0.9281609195402298, which is a quite good accuracy score for a baseline model and came to shock to us. A qualitative result we found curious was that the baseline model has a harder time accurately predicting pneumonia samples, perhaps they're less distinct and can be easily mistaken as COVID19 lungs?

Some challenges we faced was having a surprisingly high accuracy from the baseline, for now, we will keep it and hope that our deep learning techniques can outperform these results. Furthermore, because our test and train set is quite unbalanced (a lot more pneumonia images than the other 2 classes), we actually noticed that this resulted in a grossly inflated precision score for the pneumonia classifications. To fix this, we used the lower bound for number of images of a classes (i.e 116 COVID19 test images) and balanced the data sets by using that as a cap for images per class.

5 Primary Model

5.1 Architecture

The architecture of our primary model is illustrated below:

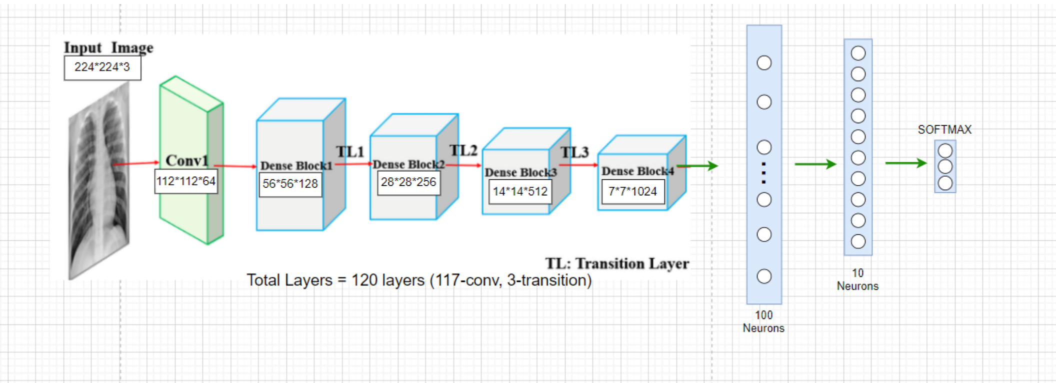
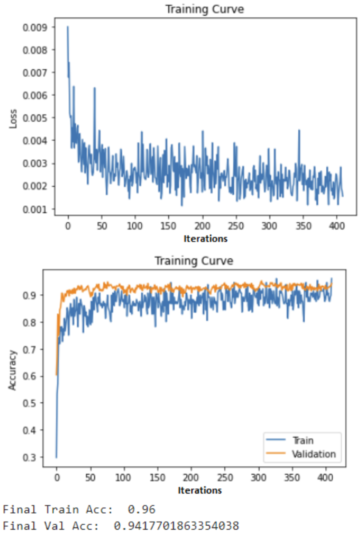


Figure 7: Our best performing architecture so far

Based on prior research, we decided to utilize DenseNet121 features as our pre-trained CNN model. We perform transfer learning, passing our images through DenseNet first, then feed the features outputted in to our own ANN. We train with mini-batching.

5.2 Current Results

Currently, our best validation accuracy's achieved with two fully connected layer with 100 neurons in first layer and 10 in second. This model achieved a validation accuracy of 94%. This model is fed a feature of shape [1024,7,7], resulting in (1024*7*7*100+100)+(100*10+10)+(10*3)+3≈5 million parameters currently. Note that DenseNet's parameters are frozen for this process and not counted. The accuracy curve is shown below.



```
Final Train Acc:  0.96
Final Val Acc:  0.9417701863354038
```

Figure 8: Our best training results so far

5.3 Qualitative Analysis

Some of the more complicated models we've tried thus far haven't yielded results as successful as the simple one above, we hypothesize the other methods are more computationally intensive and we haven't reached their convergence yet. Moving forward we seek to test techniques such as Dropout and weight decay along with deeper layers and other hyperparameter tuning.

5.4 Challenges

A huge challenge we encountered and will likely continue to deal with is the lack of computational resources we have. We saw an astronomical improvement in training speed with cuda set up. However, Google Colab temporarily bans free users from GPU runtimes when it sees too much usage.