Convolutional Neural Networks for Recognizing Covid-Induced Pneumonia in Lung X-Ray Scans

Ryan Divigalpitiya, Heather Hartley, Mohmmd Alanbary
Department of Computer Science
Western University
London, Ontario, Canada
rdivigal@uwo.ca
hhartley@uwo.ca
malanbar@uwo.ca

***Abstract*—COVID-19 has plagued our world for over a year. The AI community is continuously looking for better ways to detect pneumonia caused by COVID. In this work, we apply convolutional neural networks (CNNs) to the task of classification of COVID- induced Pneumonia. We trained a classifier to discriminate between healthy, non-COVID pneumonia infected, and COVID-induced pneumonia infected patients. Our classifiers achieve a 0.88 sensitivity and 89% accuracy on the test set.**

***Index Terms*—COVID19, X-Ray, CNN**

I. INTRODUCTION

COVID19, a current global pandemic, originated in Wuhan in late 2019 and has since spread to all corners of the world. Nearly all major economies of the world have had to go through complete or partial lockdown to mitigate the spread of the virus. This in turn has had a devastating effect on the global economy. Most commonly used COVID19 detection techniques are based on PCR-based tests which can take several hours to conduct. Due to hospitals being overwhelmed, efforts have gone underway to create faster diagnosing tools to reduce congestion in our medical system. As a result, the AI community has been sought after to explore possible ways to utilize computer vision to potentially improve both detection accuracy and speed.

Thus, our project aims to explore, build and tune a convolutional neural network for recognizing COVID-induced pneumonia in lung x-ray scans of patients. COVID can potentially be diagnosed from x-ray scans of a patient's lungs due to COVID-induced pneumonia causing inflammation and mucus buildup in the lungs. In an x-ray scan, the pneumonia will show up in the lungs as more opaque compared to a healthy patient.

We are choosing Convolutional Neural Networks (CNNs) as they have shown remarkable success in classification related tasks across multiple domains. Much of this is due to the remarkable ability of the CNNs to learn hierarchy of features to capture the essential properties of the underlying task. Starting with the simple Le-Net network for the classification of MNIST, CNNs are now able to handle image classification where the number of classes can be of the order of thousands.

Convolutional neural networks are primarily designed with three fundamental building blocks:

- 2D convolutional filters
- Max pooling
- Non-linear layers

For classification tasks there is almost always a fully connected simple neural network towards the end of the CNNs, which essentially aggregates the features extracted from the images into class labels. The simple neural network consist of two components: (1) Linear Layer and (2) Non-Linearity.

II. BACKGROUND & RELATED WORK

CNNs have been shown to be more effective at detecting chest diseases and various pathology over other known methods [1]**.** In the study conducted by Abiyev, Rahib H, and Mohammad Khaleel Sallam Ma'aitah regarding Chest Disease Detection, three types of neural networks were employed and compared; Convolutional Neural Networks, Back Propagation Neural Networks and Competitive Neural Network. It was shown that CNNs showed higher ability to generalize on large scale datasets, although they were moderately more computationally expensive. The outperformance is largely attributed to the comparative depth of a CNN and its ability to extract different level features [1].

III. DATA

Sourcing our x-ray data proved challenging for three reasons. One, x-ray scans of patient lungs can come in three different anatomical planes: Transverse, Coronal and Sagittal. After vetting through several x-ray datasets, we found most of them to have been taken via different anatomical planes, which would be very problematic when training our model. The anatomical plane of the scan must be a variable that is controlled and consistent as we do not want it to impact our model's learning. Eventually, we found a suitable dataset that only included coronal-orientated scans

(x-ray scans that were taken face-on of the patient.) Two, numerous x-ray datasets contained x-ray images that had varied cropping styles — some x-ray scans were cropped from the top-left corner, leaving significant empty space in the images, while others were cropped from different corners. Three, some datasets had different aspect ratios and dimensions throughout the x-ray scans. Building a convolutional neural net that is aspect-ratio independent is challenging, so, to make our work easier, we only chose x-ray scans that had the same aspect ratios. Eventually, we found a well-made and clean dataset that fit our criteria, sourced from Kaggle — the dataset was put together by several universities [2].

The dataset originally had four classes: healthy, COVID, bacterial pneumonia and viral pneumonia. We wanted to build a model that can differentiate between healthy, COVID and Non-COVID pneumonia. Thus, we had to combine the original bacterial and viral pneumonia classes into a singular class: Non-COVID pneumonia. The resulting data quantities for each class were as follows: Healthy: 10,000 images, COVID: 3,600 images, Non-COVID: 7,300 images.

IV. DATA-PREPROCESSING

Preprocessing of the lung x-ray images in our dataset to adjust and improve the contrast of an image will improve model performance. Image contrast adjustment is particularly useful when the background or foreground of images are either light or dark, as is the case in X-Ray images. It allows for correction of either under or overexposed images [3].

There are a few ways to approach the contrast altering of the images in the dataset:

HISTOGRAM EQUALIZATION.

Histogram equalization uses each image's histogram to alter the image's contrast. The typical end result of using histogram equalization is an increase in an image's global contrast [3]. To explain this concept simply, consider an image that is relatively “dark”, then it is true to say that the pixels of this image are confined to relatively low values [4]. An image with good contrast will have pixels that are both light and dark in all areas of the image. Furthermore, histogram equalization normalizes the image’s histogram. Over a collection of images, both darker and lighter images will be normalized and the collection of images will be as if the lighting conditions are more or less the same [4]. The project uses the cv2 implementation of histogram equalization (cv2.equalizeHist(image)).

CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION.

In histogram equalization, we only consider the global contrast of the image. In certain cases, this can result in a loss of important information contained with the image. Contrast limited adaptive histogram equalization (CLAHE), however, uses smaller blocks of pixel data and performs contrast limited histogram equalization on each of these blocks. Contrast limiting is applied to each block wherein the contrast limit is specified and any contrast exceeding this limit is “clipped” and distributed to the other bins. In OpenCV, this limit argument is specified in the argument clipLimit. A clipLimit value of 1 will return the original image with no contrast adjustments [5]. Conversely, a large clipLimit value will allow for free rein on the part of the histogram equalization. Thus, altering the clipLimit parameter between 2-3 is generally regarded as the best practice [5]. After this, histogram equalization is applied.

The below figures show a patient's lung X-Ray scans who has been diagnosed with COVID-19. Figure 1 is the original image with no preprocessing adjustments to the images contrast. Figure 2 applies histogram equalization to the image and adjusts the image’s global contrast. Figure 3 applies contrast limited adaptive histogram equalization to the image with a clip-limit of 2. Alongside each X-Ray scan (both the original and processed scans) is its respective histogram/CDF plot of the images pixel values.

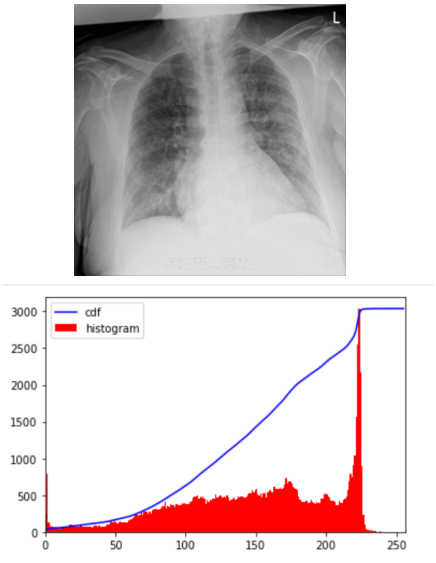

Fig. 1. Original Image and Respective Histogram/CDF Plot

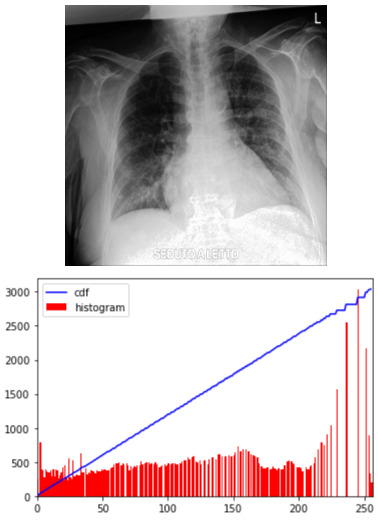

Fig. 2. Applied Histogram Equalization and Histogram/CDF Plot

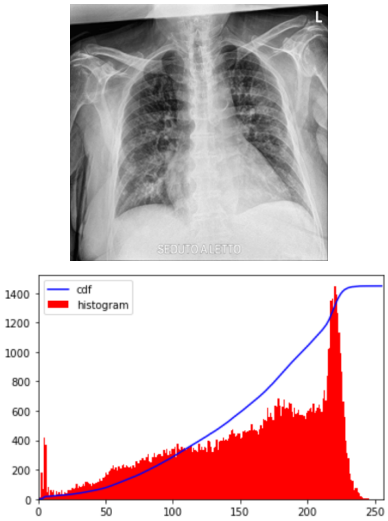

Fig. 3. Applied CLAHE and Histogram/CDF Plot

We decided to proceed with CLAHE as our pre-processing method to enhance the model's ability to learn relevant features from the dataset. A clip-limit value of 2.0 was used.

V. METHODS

RESEARCH OBJECTIVES.

Our primary objective is to build, train and tune our CNN such that it achieves both a high accuracy and a high sensitivity. The motivation for these targets are discussed in the Results section.

RESEARCH METHODOLOGY.

Our model's architecture was initially designed similarly to AlexNet's architecture and is described in Table 1. There are five convolutional layers, with Relu Activation functions and Max Pooling in-between. A dense layer acts as the networks final layer before making a prediction of one of the three classes. From there, we devised a tuning strategy to test different model architectures starting with the architecture stated in Table 1.

TABLE 1: INITIAL MODEL ARCHITECTURE

Input Image	**Conv Layer 1**	**Conv Layer 2**	**Conv Layer 3**	**Conv Layer 4**	**Conv Layer 5**	**Dense Layer**
299 x 299, 3	16 filters, 3x3	32 filters, 3x3	64 filters, 3x3	64 filters, 3x3	64 filters, 3x3	512 Neurons
Output Shape:	148, 148, 16	73, 73, 32	35, 35, 64	16, 16, 64	7, 7, 64	

Our tuning strategy composed of building six different model architectures, outlined in Table 2, each with different parameters for the filter in convolutional layer five and different neuron density in the last layer. We then experimented to determine which model performed the best according to several evaluation metrics discussed in the next section(s).

TABLE 2: MODEL PARAMETERS USED TO TUNE PERFORMANCE

Model Parameters	**Layer 5 Filter Parameters**	**Neurons in Dense Layer**
Model 1	64	256
Model 2	64	512
Model 3	64	1028
Model 4	128	256
Model 5	128	512
Model 6	128	1028

Our data was split into 90% for training and 10% for testing; we felt it was safe to allocate a higher portion for training since our dataset is very large in size. 10 epochs was used. RMSprop algorithm was used as the optimizer with a learning rate of 0.001. Finally, we used a feedback cycle consisting of evaluating each model on the test set using

accuracy, multi-classification confusion matrices, precision, recall, F1-score, sensitivity and specificity.

VI. RESULTS

As mentioned previously, accuracy, multi-classification confusion matrices, precision, recall, F1-score, sensitivity and specificity were generated for all six models in order to compare their performances. We will show all evaluation metrics only for Model One in Table 3, while giving a summary of highlighted evaluation metrics for all six models, in Table 4. Class 0 is Covid, Class 1 is Healthy, Class 2 is Non-Covid Pneumonia.

TABLE 3: MODEL 1 EVALUATION METRICS

64 Filter Parameters in Convolutional Layer 5
256 nuerons in Dense layer

Train Accuracy:	**93%**
Test Accuracy:	**89%**

Confusion Matrix:		**Actual**		
	Classes	**0**	**1**	**2**
Predicted	**0**	256	55	24
	1	10	893	57
	2	9	64	629

Classes:	**0**	**1**	**2**	**Average**	**Stdev**
Precision:	0.93	0.88	0.89	**0.90**	0.03
Recall:	0.77	0.93	0.90	**0.86**	0.09
F1-Score:	0.84	0.90	0.89	**0.88**	0.03
Sensitivity	0.77	0.93	0.90	**0.86**	0.09
Specificity	0.95	0.93	0.94	**0.94**	0.01

TABLE 4: SENSITIVITY AND TEST ACCURACY FOR ALL MODELS

	Sensitivity	**Test Acc**	
Model 1	0.86	89%	
Model 2	**0.88**	87%	
Model 3	0.77	78%	
Model 4	0.83	86%	
Model 5	0.86	88%	
Model 6	**0.88**	**89%**	← **best model**

For medical diagnosis tools, assessing false negative rates is a very high priority. Falsely classifying a patient has not having a disease when they in fact do is incredibly dangerous, so, for our model evaluation, we must pay attention to sensitivity. A higher sensitivity means a lower false negative rate. Thus, Model 6 has both the highest sensitivity and highest test accuracy amongst all our model architectures.

Our results are quite intuitive, as, if we look back at Table 2, we can see that model 6 has the most complex architecture. Having a more complex network architecture seems to allow model 6 to better learn the various complexities in the x-ray features of the patient's lungs.

VII. CONCLUSIONS & FUTURE WORK

We explored building, training and tuning CNNs to help diagnose COVID in x-rays of patient lungs. After utilizing CLAHE pre-processing and tuning the architecture of our CNN, we were able to produce a CNN model that has 0.88 sensitivity and 89% accuracy on the test set.

We have demonstrated that the use of CNNs for the aim of diagnosing COVID has great potential. Further studies would involve exploring more pre-processing techniques to bring out relevant features in the x-ray images while also experimenting with other, more complex network architectures. Transfer learning should also be explored in future work.

REFERENCES

(1) Abiyev, R. and Ma'aitah, M., 2021. Deep Convolutional Neural Networks for Chest Diseases Detection. Available at: <https://pubmed.ncbi.nlm.nih.gov/30154989/>

(2) Kaggle.com. 2021. COVID-19 Radiography Database. [online] Available at: <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database?fbclid=IwAR1xMtVl3DHi56pb8pGQe5IWDnfR_pHTvKuWmVrn-XFRap-XWyCBXn2ANLo>.

(3) En.wikipedia.org. 2021. Histogram equalization - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Histogram_equalization>.

(4) Opencv-python-tutroals.readthedocs.io. 2021. Histograms - 2: Histogram Equalization — OpenCV-Python Tutorials 1 documentation. [online] Available at: <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_equalization/py_histogram_equalization.html> .

(5) LLC, I., 2021. ImageMagick. [online] ImageMagick. Available at: <https://imagemagick.org/script/clahe.php#:~:text=clip%2Dlimit%20is%20the%20contrast,result%20in%20the%20original%20image.> .