COVision: A Novel CNN for the Differentiation of COVID-19 from Common Pulmonary Conditions

Timothy J. Mathew and Kush V. Parikh

Troy High School, MI

Abstract: With the growing amount of COVID-19 cases, especially in developing countries with limited medical resources, it's essential to accurately diagnose COVID-19 with high specificity. Due to characteristic ground-glass opacities (GGOs), present in both COVID-19 and other acute lung diseases, misdiagnosis occurs often — 26.6% of the time in manual interpretations of CT scans. Current deep learning models can identify COVID-19 but cannot distinguish it from other common lung diseases like bacterial pneumonia. COVision is a novel multi-classification convolutional neural network (CNN) that can differentiate COVID-19 from other common lung diseases, with a low false-positivity rate. COVision achieved an accuracy of 95.8%, AUROC of 0.970, and F1-score of 0.954. We found statistical significance that COVision performs better than three independent radiologists, especially on differentiating COVID-19 from pneumonia. In a controlled study comparing COVision to other state-of-the-art architectures, our novel model achieved a higher accuracy on a small training set even with a lower complexity/runtime. After training COVision with 105,000 CT scans, we analyzed our model's activation maps and found evidence that lesions in COVID-19 (specifically GGOs) presented peripherally, closer to the pleura. Out of all clinical factors tested, shortness of breath was the most indicative of a COVID-19 diagnosis. Finally, using a federated learning model, we ensembled our CNN with a pretrained neural network on clinical factors (age, symptoms, etc.) to create a comprehensive diagnostic tool. This tool was approved by a board-certified M.D. for possible adoption into a hospital. COVision is a novel, scalable, high specificity tool that provides immediate diagnosis of lung diseases using CT scans and/or clinical factors.

Abbreviations: area under the receiver operating characteristic (AUROC), Cascading Style Sheets (CSS), categorical cross-entropy (CCE), Chest X-Rays (CXRs), clinical factor neural network (CFNN), comma-separated values (CSV), Compute Unified Device Architecture (CUDA), computed tomography scan (CT), convolutional neural network (CNN), false negative (FN), false positive (FP), fully connected neural network (FCNN), gradient weighted class activation mapping (Grad-CAMs), graphical processing unit (GPU), ground-glass opacities (GGOs), HyperText Markup Language (HTML) NVIDIA CUDA Deep Neural Network (cuDNN), optical coherence tomography scans (OCT), Rectified Linear Unit (ReLU), reverse transcription-polymerase chain reaction (RT-PCR), simple random sample (SRS), true negatives (TN), true positives (TP), World Health Organization (WHO)

1. INTRODUCTION

1.1 Background

The outbreak of severe acute respiratory syndrome coronavirus-2 (SARS-CoV-2) and its associated disease COVID-19 has led to a global pandemic. As of March 31st, 2022, there have been over 486 million COVID-19 cases worldwide, claiming an estimated 6.14 million lives according to the World Health Organization (WHO)[1]. COVID-19 infects the lungs, specifically the alveolar type II cells, typically resulting in complications like pneumonia[2]. Currently, RT-PCR remains the gold standard for COVID-19 diagnosis; however, due to limited sensitivity of 89.9%[3], and a wait time of at least 48 hours for results, the need for quicker, and more accurate diagnosis is imperative[4]. This is especially the case when patients present to the hospital with severe respiratory disease that could be COVID-19, but also a condition with similar presentations such as bacterial pneumonia (Figure 1)[5], pulmonary edema, or sepsis. Because of the similarity in presentation of these pulmonary conditions, it is often difficult to form an accurate diagnosis with CT scans alone, leading to a high rate of misdiagnosis[6]. To this end, a high specificity deep learning model that can quickly and accurately diagnose and separate COVID-19 CT Scans from other lung conditions, complete with a public application, has yet to be developed.

1.2 Disproportionate Effect of COVID-19

The disparity in the COVID-19 healthcare response between developing and developed countries is staggering. According to the World Bank, high and high-intermediate countries have a higher physicians per capita and nurses per capita when compared to low and low-intermediate income countries[7]. Factors such as slow economic growth in developing countries and the migration of healthcare workers from developing to developed countries are the primary reasons attributed to the lack of healthcare professionals in developing nations[7]. The shortage of healthcare workers in the low and low-intermediate countries has led to greater work hours per week and higher rates of burnout[8]. These issues have only been exacerbated due to the COVID-19 pandemic leading to overburdened medical systems. Using digital technology and automation in healthcare, particularly in low income nations, has great potential to ease the burden due to the COVID-19 pandemic on these nations' already crumbling medical infrastructure.

1.3 Deep Learning

New developments in deep learning have led to innovative potential diagnostic applications. Deep learning allows for the extraction of subtle quantitative features in datasets allowing for analysis of complex patterns in the training data, leading to the

possibility of creating automated high-accuracy diagnosis models using medical scans in radionomics[9]. The convolutional neural network's (CNN) ability to use historical recall of data, and the use of nonlinear systems (as opposed to commonly used linear systems) allows for more accurate classification. In the past, CNNs have shown general usability in diagnosing retinal conditions using optical coherence tomography scans (OCT)[10].

1.4 Existing Works

SARS-Net[11] is one of many deep learning models developed to aid with COVID-19 diagnosis. While this model is able to achieve an accuracy of 97.6% in identifying COVID-19 from Chest X-Rays (CXRs), this model fails to differentiate COVID-19 from other common pulmonary conditions such as bacterial pneumonia leading to a low specificity. Specificity is a measure of how well a model can identify individuals who do not have a disease and can correctly identify what condition(s) an individual might have instead. For effective use in a clinical setting, and for triaging of patients, models that detect COVID-19 from medical images like CXRs or CT Scans must have a high specificity.

Currently many state-of-the-art architectures such as ResNet152[12], VGG19[13], and InceptionV3[14] would be used as frameworks for developing models. However, the complexity of these models often leads to overfitting and an extremely high runtime. In our research, we aimed to design simpler models, with a much simpler architecture complexity and lower runtimes, while still achieving a high accuracy on testing sets.

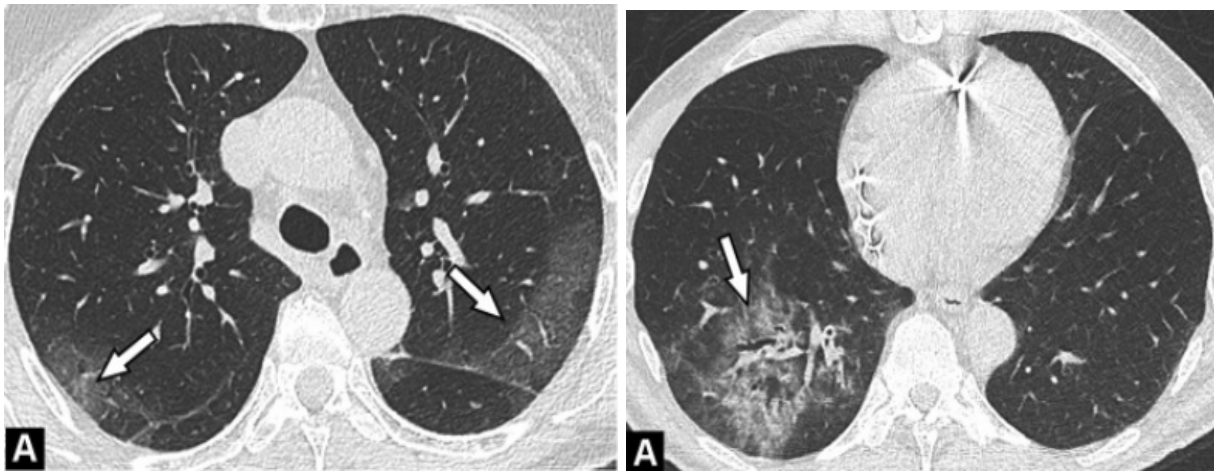

Figure 1. Image of two patients with ground glass opacities on a coronal CT scan of the lungs; 71 year old with COVID-19 (left) and 45 year old with non-viral pneumonia (right)[5].

Source: Hani, C., Trieu, N. H., Saab, I., Dangeard, S., Bennani, S., Chassagnon, G., & Revel, M.-P. (2020). COVID-19 pneumonia: a review of typical CT findings and differential diagnosis. Diagnostic & Interventional Imaging. https://doi.org/10.1016/j.diii.2020.03.014

2. METHODS

2.1 CT Scan Data Augmentation and Preprocessing

194,922 isolated CT slices for 3475 patients were obtained from the CC-CCII dataset[15]. The slices were split into 80:20 ratio of training images to testing images. To standardize the images, all the images were resized into a size of {512, 512, 1} through Lanczos3 interpolation. Lanczos resampling rescales the images by passing the pixels in the image through an algorithm based on *sinc* functions. This type of interpolation minimizes the aliasing, which is crucial for the model to develop accurate patterns. Layers of augmentation were then applied othe training images to increase the diversity of the data. By altering the brightness, saturation, rotation of the images and by adding Guassian noise, the model prevents overfitting by reducing bias in the training data.

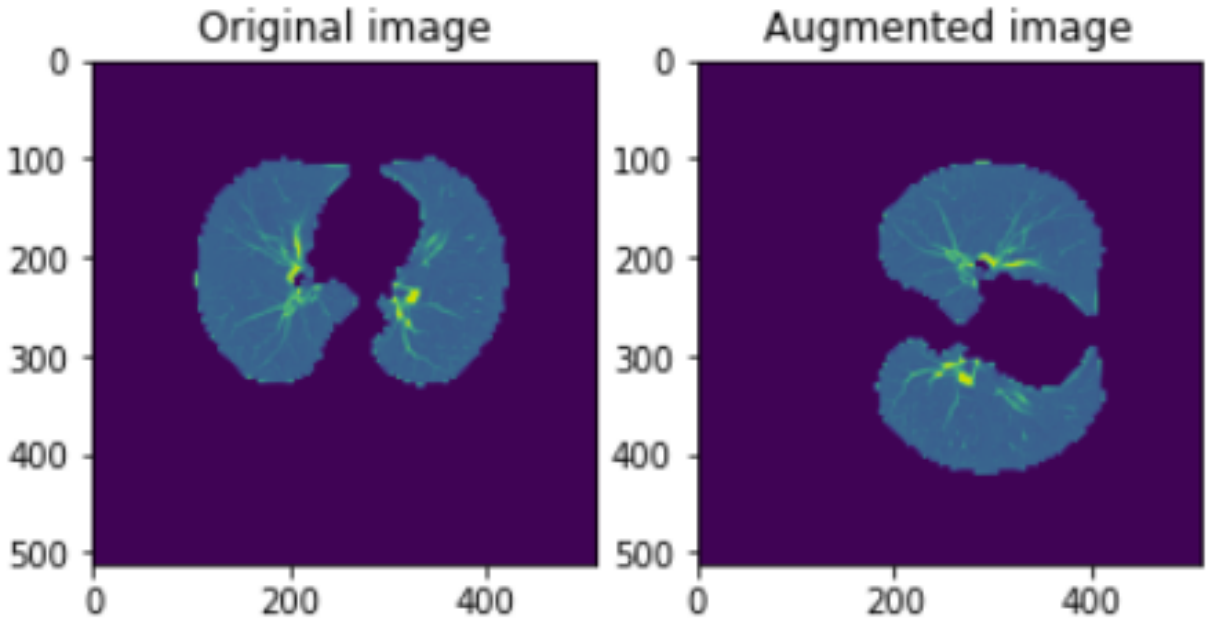

Figure 2. Example of original vs augmented CT slice (rotation filter). Source: [Authors]

2.2 Proposed Novel Convolutional Neural Network

The augmented images are passed into the first layer with size 512 x 512 x 1. The first 2 dimensional convolutional layer contains 64 3x3 Gaussian kernels (Figure 3, left) with a stride of 1x1 because of its edge detection ability. We used this kernel size (3x3) because it is symmetric around the center, and extracts a large amount of details from the image. While this does increase the computational expense, the difference in computation from a 3x3 filter to filters of greater sizes is marginal. These filters extract features from the

images by applying convolution operations to create a feature map. The feature maps are transformed by the Rectified Linear Unit (ReLU) activation function which prevents exponential growth in the computation by assigning 0 to negative input values, thereby activating less neurons in the feature map by zeroing values that do not contain any information. Spacial dimensions are then reduced using a 2x2 max-pooling (Figure 3, right) filter which significantly reduces the computational cost by reducing the number of parameters to learn. Lesions such as GGOs, crazy-paving patterns, and consolidation in the lungs all show up on a CT Scan as brighter pixels. Brighter pixels have grayscale values closer to 1 while darker pixels have grayscale values closer to 0. This is why we use maximum pooling instead of minimum pooling because on CT Scans, the maximum values (i.e. the brightest pixels) contain the most relevant information about the image needed for classification of lung diseases. The resulting feature maps contain high-level features which are then classified by a multilayer perceptron network after being flattened. A *Softmax* activation is used to normalize the output from the last fully connected layer into a multinomial probability distribution over K classes. Here, K = 3 for COVID-19, bacterial pneumonia, and healthy slices. In total, the CNN contains 6,542,531 trainable parameters.

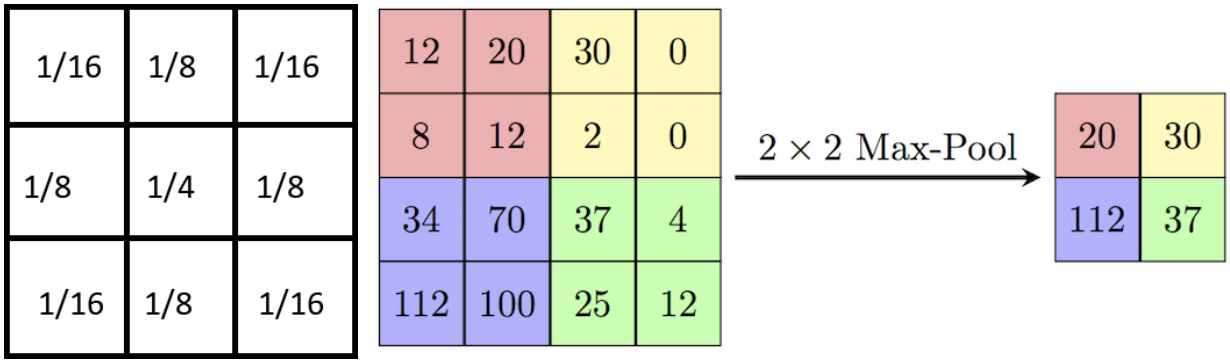

Figure 3. Discrete approximation of a 3x3 Gaussian kernel (left). The 9 pixels that the kernel is applied to every stride are multiplied by certain factors which are shown. Visualization of a 2x2 Max-Pool function (right). For every 2x2 section of pixels in which the filter is applied, the maximum value is output, reducing feature size by a factor of 4.

Source: *Papers with Code - Max Pooling Explained*. (n.d.). https://paperswithcode.com

2.2.1 Minimizing Complexity

Between the input and output layers of a neural network, a series of hidden layers are used to identify various patterns within the training data. The training accuracy of a CNN will generally increase with more hidden layers, along with the computation and complexity of the model. An overly complex model will often overfit because it learns the patterns in the training data so well that it isn't able to extrapolate to testing data. This

means there is a tradeoff between the complexity and the accuracy the model achieves on testing data. Current state-of-the-art models VGG19, InceptionV3, and ResNet152 have 19, 48, and 152 layers respectively. These large numbers of layers significantly increase how much the model overfits because the model is too complex. This complexity also increases the time to train the model because of the substantial amount of computation that comes along with additional increased layers. On computing systems with lower computation power, these models may be untrainable in certain scenarios due to the immense amount of computation required. With this in mind, COVision was designed to classify image features with just 6 hidden layers. Together with the input and output layers, COVision minimizes unnecessary computation and complexity with just 8 layers in total.

2.2.2 Dropout Layers

After choosing the number of hidden layers in our novel architecture, we further increased accuracy and prevented overfitting by implementing regularization through dropout layers. Dropout layers randomly set some of the outputs of a certain layer to 0. The proportion of outputs that are dropped out is based on the dropout factor p such that the probability an output in a certain layer is dropped is $1 - p$. We placed dropout layers after the 1st and 2nd max-pool layers and after the 1st and 2nd dense layers (Figure 4).

Standard convention is to set dropout $p = 0.5$ for fully connected (dense) layers and $p = 0.8$ *or* 0.9 for convolutional layers, however this technique is arbitrary and is not generalizable to every CNN. Using *GridSearchCV* from *sklearn* library, we use grid searching to test dropout factors between 0.1 and 0.9 (increment = 0.1) in combination for all four dropout layers. The following set of dropout factors achieved the highest accuracy: 0.6 for between the convolutional layers and 0.7 for between the dense layers.

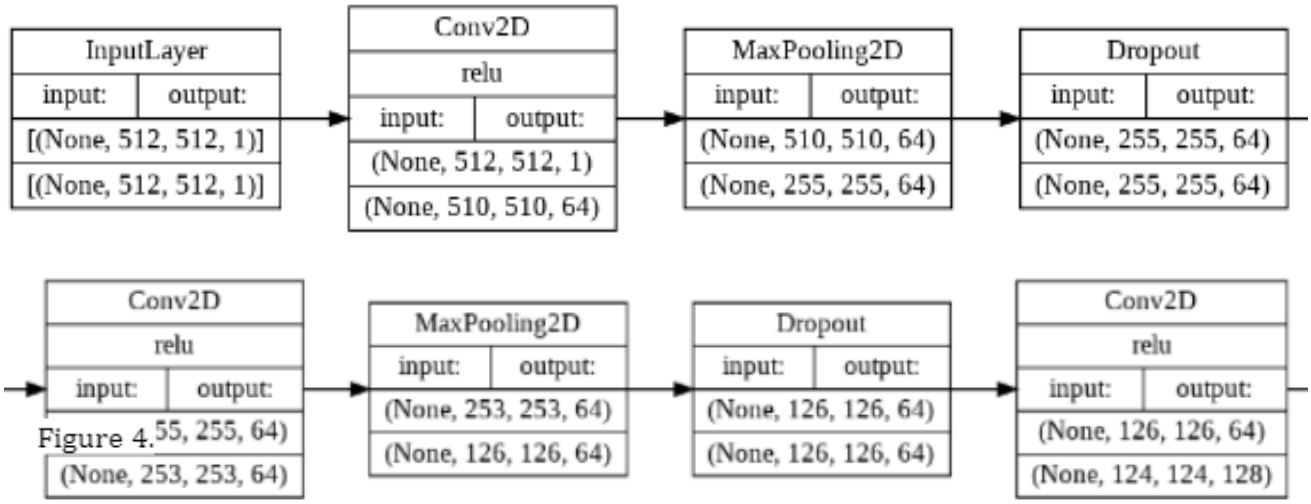

Figure 4.

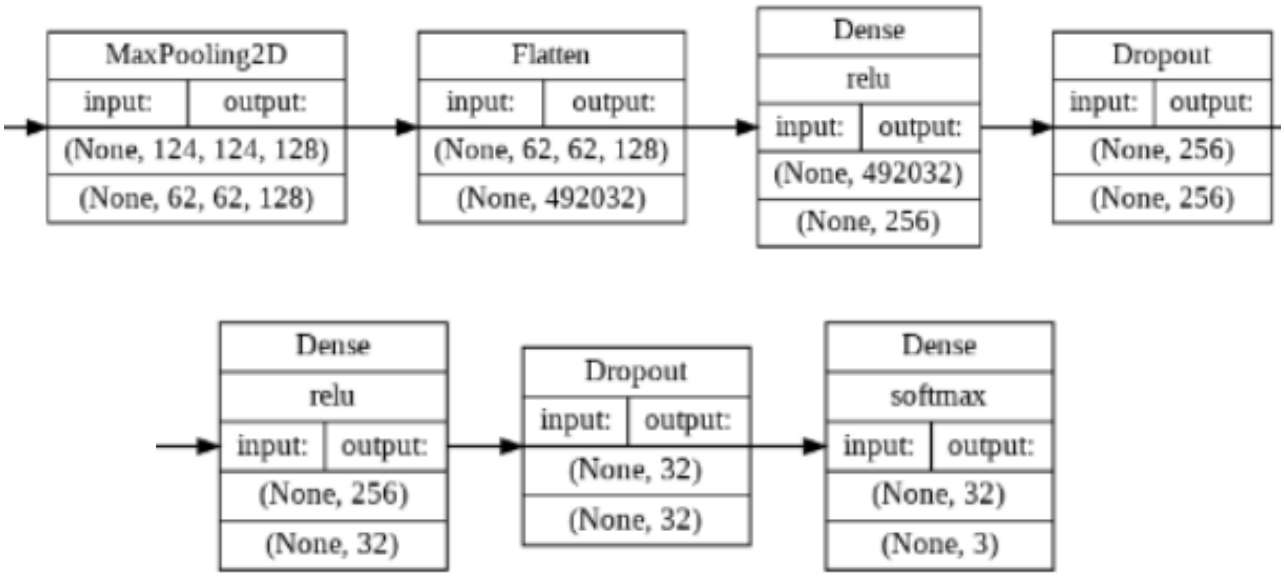

Figure 4. CNN Architecture; Input & output shape for each layer shown. Source: [Authors]

2.3 Training Novel Convolutional Neural Network

COVision was trained using a stratified random sample of 105,000 isolated CT slices taken from our training set (Section 2.1). We used 35,000 slices for COVID-19, pneumonia , and healthy (control). We trained our CNN on a NVIDIA GeForce 3090 GPU for 250 epochs by using CUDA, which enabled the GPU to be used for general purpose computing. The Tensorflow operations performed on the NVIDIA GPU were accelerated with the *cuDNN* library. All Python scripts were run from the Linux Command Line.

2.3.1 Initializing Weights

Before training, weights are assigned to the untrained model. Note that weights are not initialized to 0 because the derivatives in gradient descent would remain the same, which prevents the weights from being optimized and the model from learning. Instead, random initialization is used to break this symmetry. COVision initializes its weights using a Glorot (Xavier) Uniform Initializer[16] because of its ability to maintain variance across layers, which prevents the gradients from exploding or vanishing. The weights for each layer are chosen by randomly selecting samples from the range on a uniform distribution. The range is calculated using the number of input and output neurons for every layer.

$$Uniform\ Distribution\ of\ [-\ x,\ x]\ where\ x\ =\ \sqrt{\frac{6}{inputs + outputs}}$$

2.3.2 Loss Function

The model's weights are modified slightly during every epoch during training to minimize the loss, which is a metric to evaluate how well the model's predictions align with the ground truth. COVision's loss is calculated by Categorical Cross Entropy[17]:

$$Loss_{CCE} = -\sum_{i=1}^{n} t_i * log_2(p_i)$$

This loss function first takes the model's predictions and applies a *Softmax* activation to form a probability distribution (p_i). The distance between this predicted probability distribution and the ground truth values (t_i) is calculated by cross-entropy and is penalized logarithmically so that large differences output a value of 1 while small differences output a value of 0. Specifically, the logarithm (base 2) of this distribution is multiplied with the distribution of the ground truth label for all classes (n). The categorical cross entropy of the system is calculated by summing all of these products to form a quantitative measurement of the uncertainty in the system, or lack of order in the system. A low categorical cross entropy closer to 0 indicates the current set of weights are able to classify the training CT Scans with a high accuracy. A high categorical cross entropy closer to 1 indicates the current set of weight classifies the training CT Scans with a low accuracy.

2.3.3 Adam's Optimizer

Through each epoch of training, the categorical cross entropy is minimized by an optimizer by adjusting the current weights of the model. COVision uses Adam's Optimizer[17] as an algorithm to minimize its categorical cross entropy loss because of its use of momentum and a non constant learning rate. Momentum allows the optimizer to overcome valleys caused by noise in the loss gradient when converging to the minima. The algorithm takes steps towards the local minimum by using the derivative of the loss function. The size of these steps are determined by the learning rate. The initial learning rate is η. Specifically, a small learning rate may require many steps until reaching the minimum loss while a large learning rate may overshoot past the minimum.

$$m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * G_t \qquad v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * G_t^2$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\frac{v_t}{1-\beta_2^{\,t}}+\varepsilon}} * \frac{m_t}{1-\beta_1^{\,t}}$$

Adam's uses an adaptive learning rate based on adaptive moment estimation. The optimizer computes the moving averages of gradient (G_t) and gradient squared (G_t^2) to estimate the moments mean (m) and uncentered variance (v) respectively. The hyperparameters for these computations were tuned using a grid-search method for COVison. Using *GridSearchCV* from the *sklearn* library in Python, a cross validation process is performed where a metric for different portions of the data are averaged to estimate the performance. This process was used to tune the initial learning rate (η), beta 1 (β1), beta 2 (β2) for COVision with root mean squared error as the metric. The hyper- parameters were tuned by a factor of 10 from a range of 0.1 to 0.0001 for η and 0.9 to 0.9999 for β1 and β2. The combination of hyperparameters that achieved the lowest root mean squared error is summarized in Table 1. The same process for grid searching the best hyperparameters is used in Section 2.6 for the clinical factor neural network (CFNN).

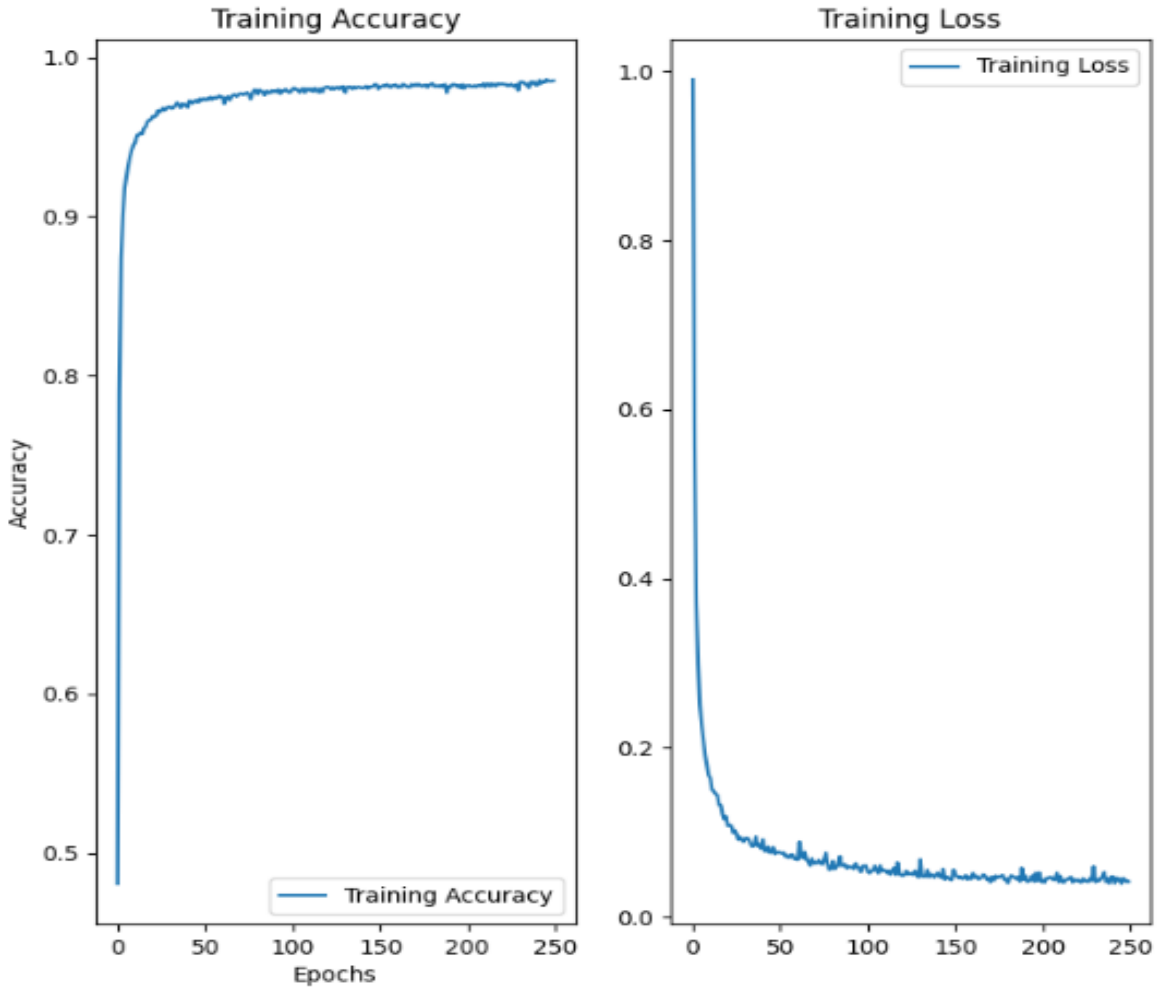

Figure 5. Graph displays accuracy of the CNN on training data for each epoch (left) and value of categorical cross-entropy loss function for each epoch (right). Source: [Authors]

2.4 Clinical Factors Dataset Oversampling

We used the Khorshid COVID Cohort (KCC)[18], and the Israeli Ministry of Health public health database[19] to construct a custom dataset of 7 clinical factors (shortness of breath, cough, headache, fever, sore throat, age, and gender). Combined, we compiled clinical factors for 30 patients with COVID-19, 30 patients with pneumonia, and 125,882 healthy patients. Training a model on this dataset would result in an imbalance classification problem because of the skewness in distribution over the three classes. To address this imbalance, the data was resampled using the *Imbalanced-Learn* library in Python. The majority class of healthy patients was undersampled so that 12,000 sets of clinical factors were randomly selected. Both minority classes of patients with COVID-19 and patients with pneumonia were oversampled through random duplication so that 11,970 sets of clinical factors were added to the original 30 sets for both classes. After applying oversampling and undersampling to the three classes, the complete dataset had 36,000 sets of clinical factors equally distributed which was split 80:20 into a training/testing set.

2.5 Clinical Factors Neural Network (CFNN)

In addition to CT Scans, a patient's clinical factors can serve as a means of differentiating whether a patient has COVID-19 or pneumonia. We designed this secondary neural network called the clinical factors neural network (CFNN) to work in conjunction with our CNN (for CT Scans) designed and trained in Sections 2.2 and 2.3 respectively. Adding another neural network to the COVision framework increases the variation during training, which consequently decreases the spread of predictions and the overall bias. The ensembling process to combine the CFNN and the CNN is described in Section 4.1.

Our CFNN is a fully connected neural network (FCNN), or multilayer perceptron neural network, with 6 fully connected (dense) layers. This means that every neuron in a specific layer is connected to every neuron in the following layer. These layers can have a varying amount of neurons, but the standard convention is to keep the amounts at powers of 2 (32, 64, 128). The output layer has a size of 3 neurons in our model, which are the 3 classes the images are categorized into. The large amount of connections increases the complexity and computation time, so we added a dropout layer for regularization after the first 3 dense layers to reduce overfitting. The dropout factor was tuned to $p = 0.5$ using the same grid-searching method in Section 2.2.2. *ReLU* was used as the activation function in all the hidden layers to prevent exponential growth in computation, and *Softmax* was used in the final layer to create the probability distribution over the 3 classes: COVID-19, pneumonia, and healthy. In total, there are 60,099 trainable parameters in our CFNN.

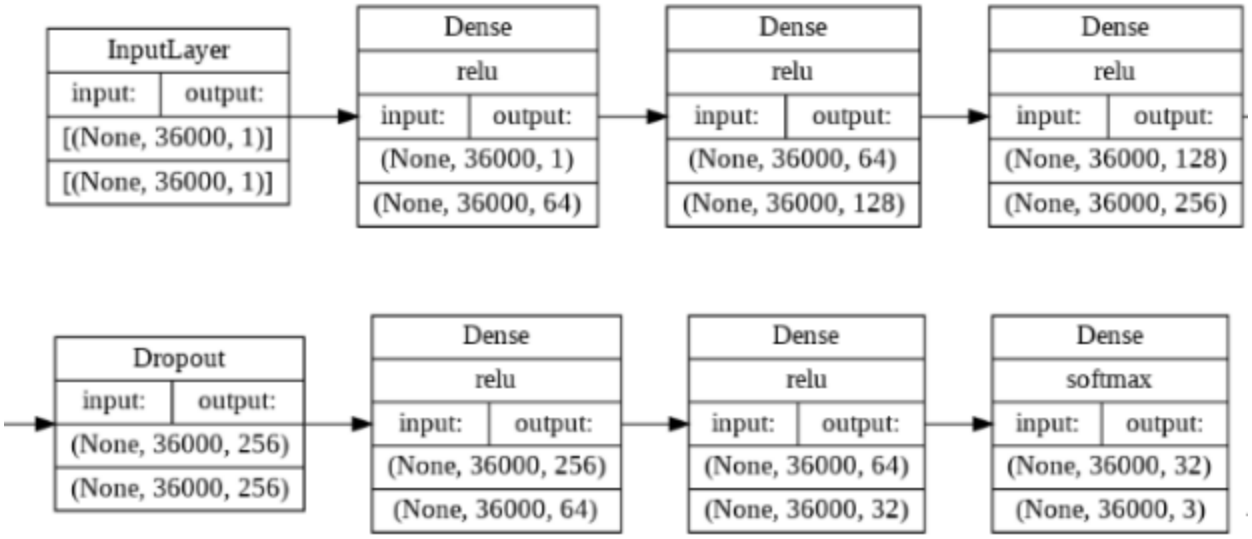

Figure 6 CFNN architecture; Input & output shape for each layer shown. Source: [Authors]

2.6 Training CFNN

We preprocessed our clinical factors training data (Section 2.4). In total we trained our model for 7 clinical factors: shortness of breath, cough, headache, fever, sore throat, age, and gender. After processing our training data, we trained our CFNN on a NVIDIA GeForce 3090 GPU using CUDA and cuDNN. We utilized *early stopping* in our training, which is a regularization method in which the amount of epochs is decreased to minimize overfitting. Both the accuracy and loss of the model began to stabilize by 40 epochs so we did not continue training our model past 50 epochs. The weights in our CFNN were initialized using a Glorot Uniform Initializer (Section 2.3.1) and the Categorical Cross Entropy loss function (Section 2.3.2). Adam's Optimizer was used to optimize the weights to minimize the Categorical Cross Entropy loss function, thereby achieving maximum accuracy. We used grid searching to choose the best hyperparameters for Adam's Optimizer (Using method described in Section 2.3.3). The optimal chosen hyper parameters are summarized in Table 1. All Python scripts were run from the Linux Command line. The network reached a maximum accuracy of 92% and a loss of 0.12.

Hyperparamater	Initial Learning Rate (η)	Beta 1 ($\beta 1$)	Beta 2 ($\beta 2$)	epsilon (ε)
CNN (CT Scans)	0.001	0.9	0.999	10^{-8}
CFNN (Clinical Factors)	0.01	0.99	0.999	10^{-8}

Table 1. CNN and CFNN Adam's optimizer hyperparameter choices. Source: [Authors]

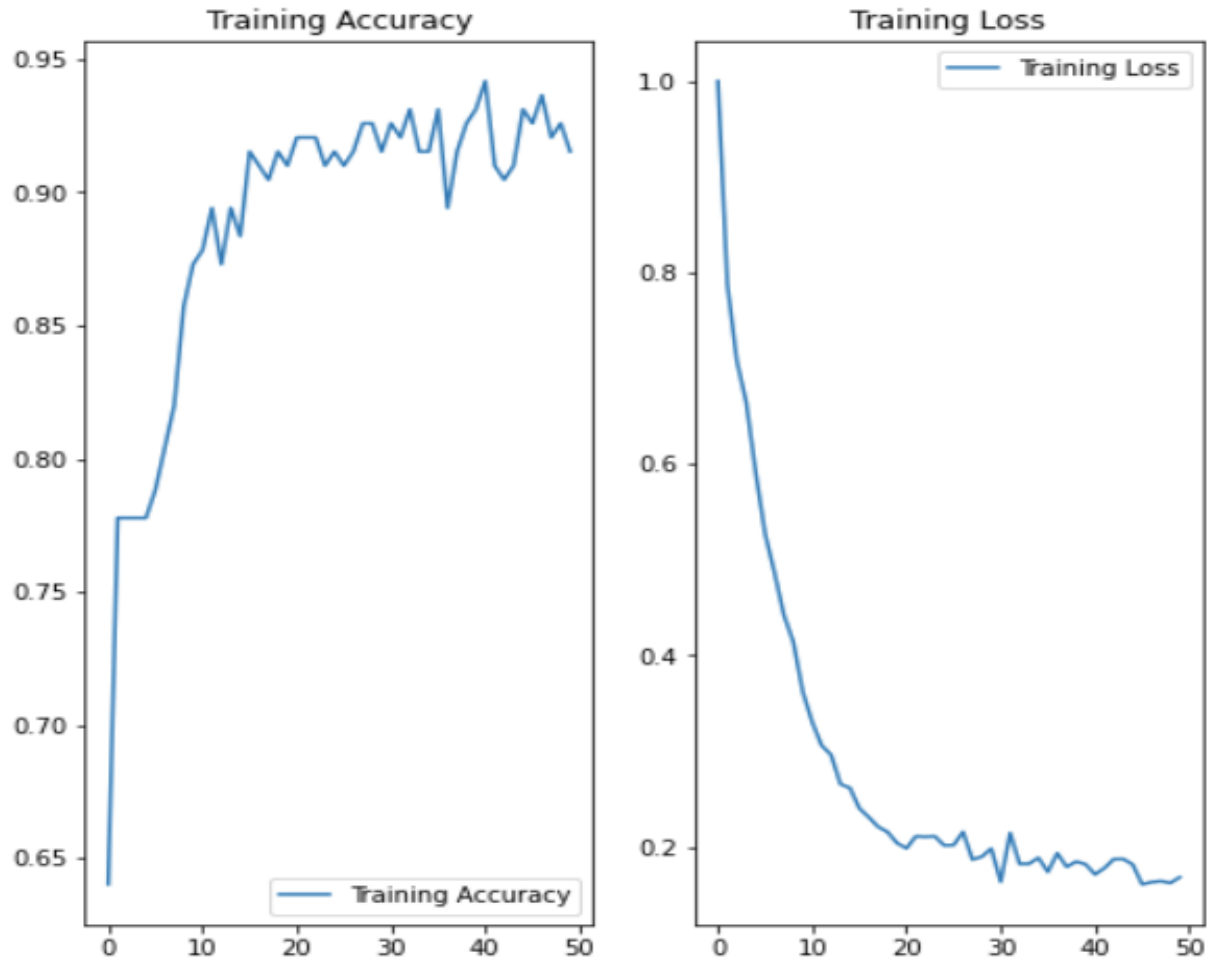

Figure 7. Graph displays accuracy of the CFNN on training data for each epoch (left) and value of categorical cross-entropy loss function for each epoch (right). Source: [Authors]

	Total	Condition	Training	Testing
CT Scans	**194,922 CT Slices in the initial dataset.** (Random samples of testing and training sets were then taken)	**COVID-19**	35000	5638
		Pneumonia	35000	7254
		Healthy	35000	12766
Clinical Factors	**36,000 Sets (After oversampling and undersampling)**	**COVID-19**	9600	2400
		Pneumonia	9600	2400
		Healthy	9600	2400

Table 2. Breakdown of all data used for testing and training of COVision. Source: [Authors]

3. RESULTS

3.1 CNN Testing

To test our trained novel CNN, we took an SRS of 25,658 isolated CT slices from our testing set (Section 2.1). The breakdown of the testing data are as follows: 12766 healthy, 7254 pneumonia, and 5638 COVID-19. Note that per design, none of the slices used for testing were a part of the training set. Results are summarized in the confusion matrix:

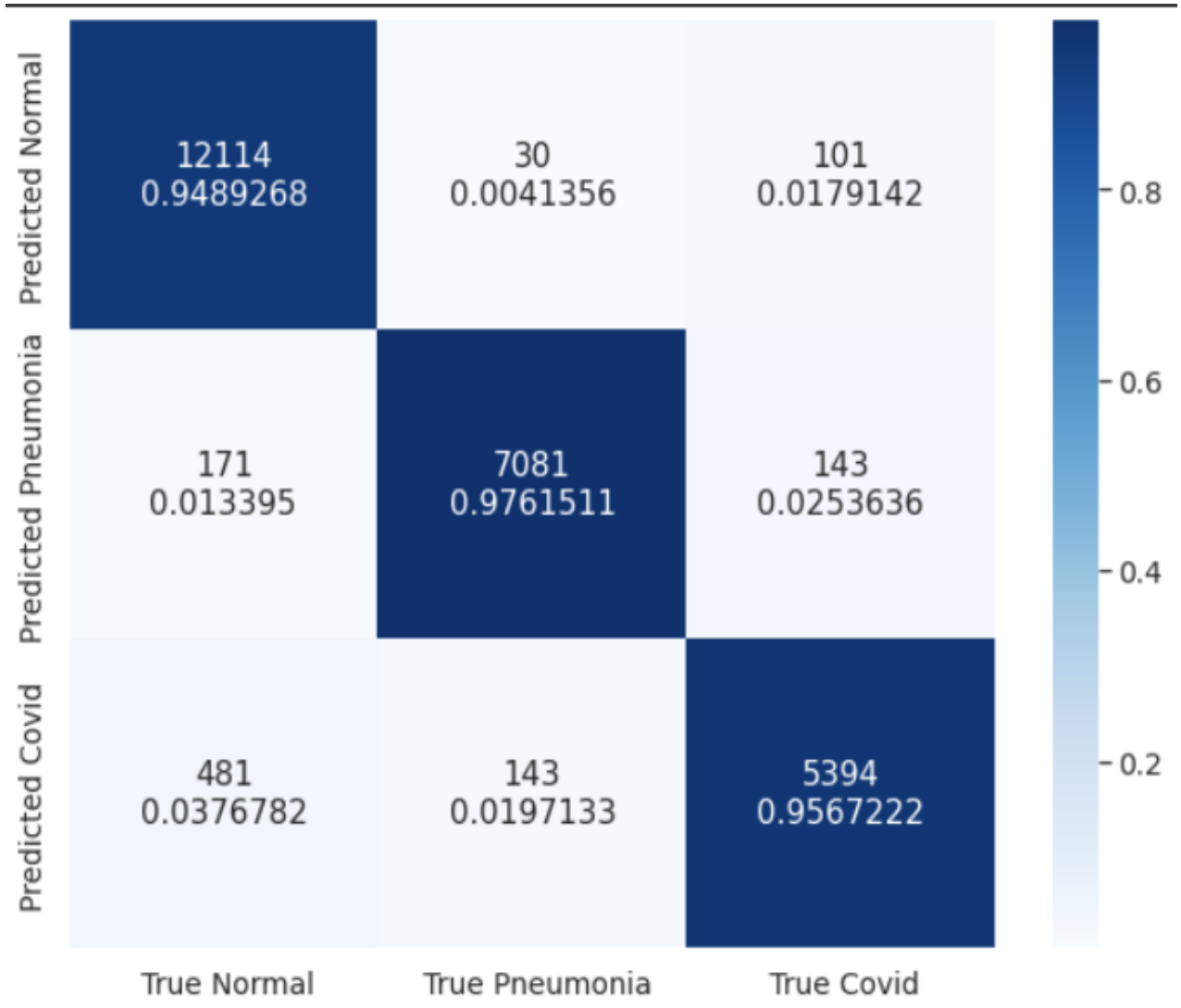

Figure 8. Confusion matrix comparing the true labels for the 25,658 CT scans and the predicted labels by our novel CNN. Proportion of images labeled accurately expressed as a decimal from 0-1. The higher the accuracy, the darker the labeled box. Source: [Authors]

For this multi-classification problem, we calculate TP, FP, TN, FN by using the "one vs all" method. For example, to calculate the FP, we calculate the FP for all three classes - true COVID, true pneumonia, and true healthy - and then take the average of all 3 values to determine the final combined FP. The following formulas are used to calculate metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \qquad Specificity\ (True\ Positive\ Rate) = \frac{TN}{TN + FP}$$

COVision achieved an accuracy of 95.8% (0.958) and specificity of 0.980 on the testing set.

AUROC (area under receiver operating characteristic) is a graphical plot that illustrates the diagnostic ability of a binary classifier system based on how well the model differentiates between different classes. AUROCs closer to 1 indicate greater separation for the three different classes (COVID, pneumonia, healthy). As our CNN differentiates between three classes and not two, we use the "one vs all" method to calculate the AUROC. To determine the combined AUROC, we took the average of AUROC for all three classes (COVID, pneumonia, and healthy). We plotted AUROC on 1-sensitivity versus specificity.

$$1 - Sensitivity\ (False\ Positive\ Rate) = 1 - \frac{TP}{TP + FN}$$

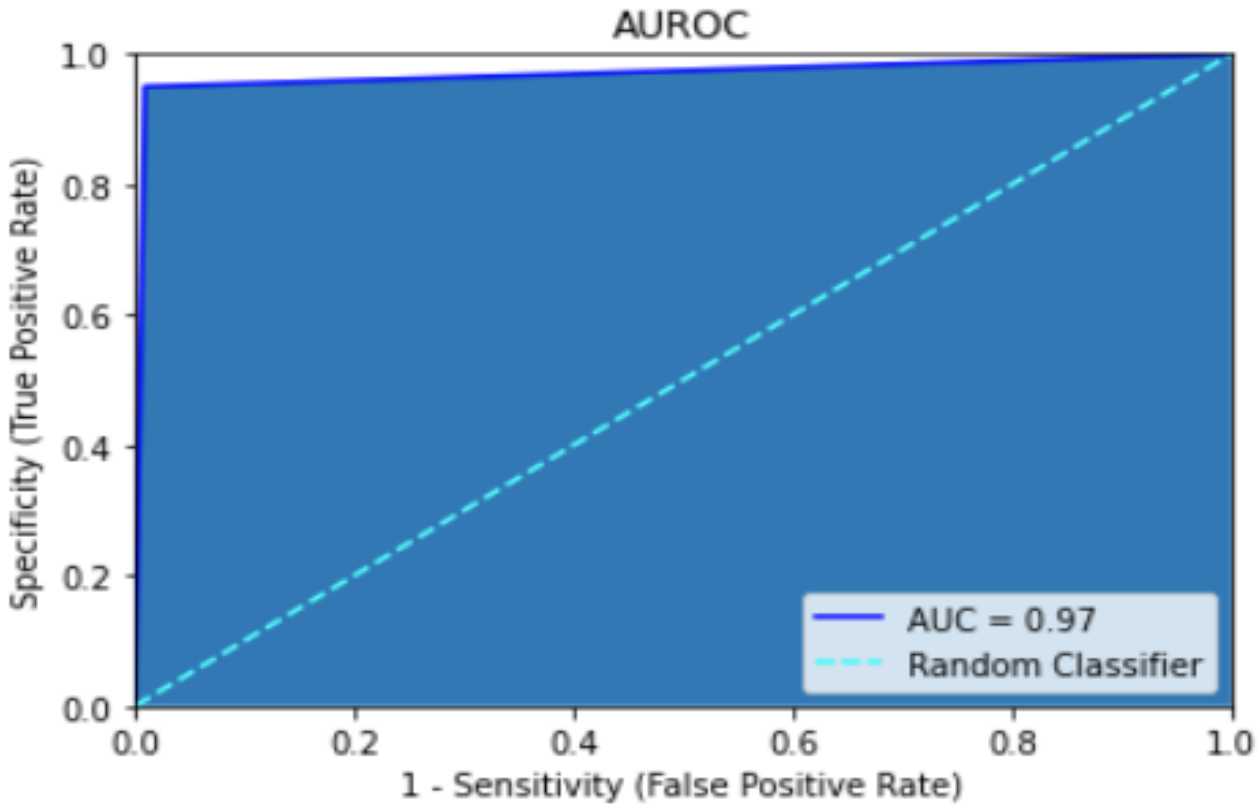

Figure 9. Visualization of COVision's AUROC versus a random classifier. Source: [Authors]

3.2 Comparison Against State-of-the-Art Architectures

We compared the performance of COVision and other state-of-the-art machine learning architectures that are used professionally such as ResNet152, VGG19, and InceptionV3. We specifically chose these models of these specific architectures to test against as they have been used in the past for other medical classification models for retinal disease and breast cancer. We took a simple random sample (SRS) of 2500 images from our training set and a simple random sample of 500 images from our testing set. We used these same images to train and test all 4 models. We trained all 4 models on the NVIDIA GeForce RTX 3090 GPU on 250 epochs. For all 4 models, we trained using Adam's Optimizer using the following hyperparameters: $\eta = 0.001$, $\beta 1 = 0.9$, $\beta 2 = 0.999$, and $\varepsilon = 10^{-8}$

We then tested all 4 models on our 500 testing images and then measured the following metrics for each model: accuracy, F1 Score, true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), and loss from the categorical cross-entropy loss function on the final epoch. To quantify complexity of each model, we analyzed the number of deep layers (convolutional, max pooling, fully connected, SoftMax), because layers are indicative of the runtime of the model when training (i.e. A model with 152 layers would take a significantly longer time to train than a model with 8 layers).

$$Recall = \frac{TP}{TP + FN} \quad Precision = \frac{TP}{TP + FP} \quad F1\ Score = \frac{2 \times precision \times recall}{precision + recall}$$

Architecture	Accuracy	F1	TP (true +)	FP (false +)	TN (true -)	FN (false -)	Loss (Cost)	Deep Layers
COVision	**0.858**	**0.878**	**306**	**33**	209	52	0.189	**8**
VGG19	0.823	0.842	282	57	212	49	**0.107**	19
InceptionV3	0.845	0.861	289	50	**218**	**43**	0.145	48
ResNet 152	0.818	0.839	284	55	207	54	0.139	152

Table 3. Comparison of accuracy, TP, FP, TN, FN, F1-Score, loss (categorical cross-entropy) ,and number of deep layers for COVision, VGG19, InceptionV3, and ResNet152 on a training size of 1000 CT slices and a testing size of 500 CT slices. Source: [Authors]

We find that COVision has a higher accuracy and F1 Score, and also had the lowest number of false positives on a limited training and testing sample while using only 8 deep layers. Other models such as VGG19 have a lower loss and InceptionV3 has the highest

number of true negatives and the lowest number of false negatives. The high true positivity and low false positivity rate indicates that COVision has a strong ability to differentiate between COVID-19 and bacterial pneumonia with a high specificity. COVision's simple structure is able to prevent overfitting unlike other professional models.

3.3 Comparison Against Radiologists

We performed a two sample z-test to determine if there was any statistical significance that COVision outperformed three independent board-certified radiologists with at least 5 years of clinical experience. We took an SRS of 297 images from our testing set and asked three radiologists to blindly classify CT scans as either COVID-19, Bacterial Pneumonia, or Healthy. Radiologist 1 classified 97 images, Radiologist 2 classified 150 images, Radiologist 3 classified 88 images. The results are visualized in Figure 10.

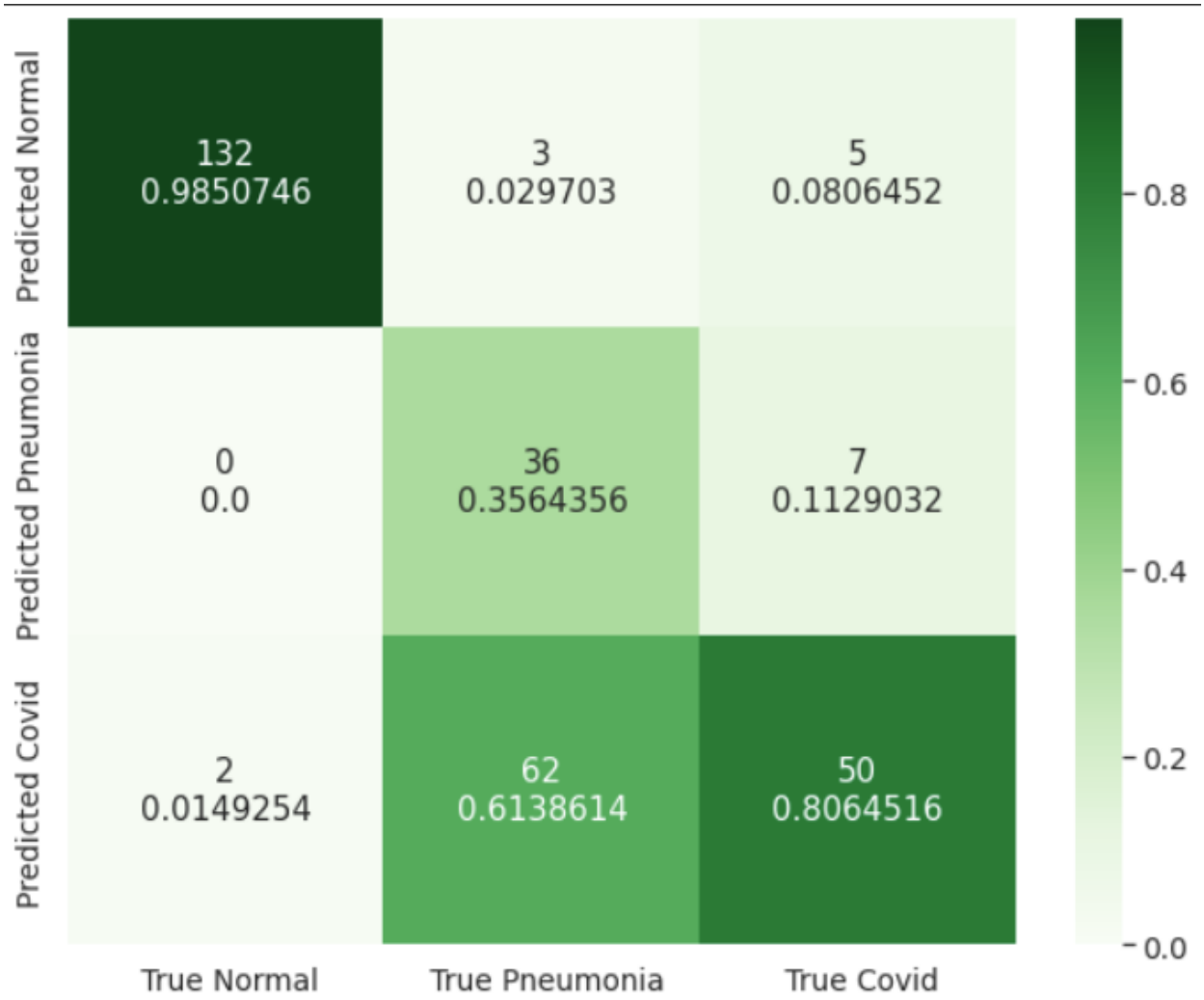

Figure 10. Confusion matrix comparing the true labels for the 297 CT scans and the predicted labels by 3 radiologists. Radiologists accuracy was 73.4%. Source: [Authors]

We performed the two sample z-test at a significance level of $\alpha = 0.05$. COVision had an accuracy of p1 = 0.958 on a testing sample size of n1 = 25658, and the radiologists had an accuracy of p2 = 0.734 on a testing sample size of n2 = 297. Our null hypothesis was that the accuracy of COVision is equal to accuracy of the three radiologists ($p1 = p2$). Our alternate hypothesis was the accuracy of COVision is greater than accuracy of three radiologists ($p1 > p2$). All conditions were met for performing the test as both samples were random (used the *sample* function from the *random* library in Python). All calculations were computed using the *statsmodels* library in Python.

$$p0 = \frac{x1 + x2}{n1 + n2} = \frac{24589 + 218}{25658 + 297} = 0.956 \quad z = \frac{p1 - p2}{\sqrt{p0 \times (1 - p0) \times (\frac{1}{n1} + \frac{1}{n2})}}$$

$$z = \frac{0.958 - 0.734}{\sqrt{0.956 \times (1 - 0.956) \times (\frac{1}{25658} + \frac{1}{297})}} = 18.66 \qquad p(z \geq 18.66) \simeq 0$$

Since the p-value of approx. 0 is less than the significance level of 0.05, there is significant evidence to reject the null hypothesis. Specifically, there is significant evidence that COVision is more accurate than the three radiologists in classifying chest CT scans as COVID-19, bacterial pneumonia, or healthy. When analyzing the confusion matrices (Figures 8 and 10), we find that COVision can differentiate COVID-19 from pneumonia with 97.8% accuracy while the three trained radiologists do this with 55.5% accuracy.

3.4 Gradient-Weighted Class Activation Mapping for CNN

To visualize the weights of the trained CNN we created Gradient-Weighted Class Activation Mapping (Grad-CAMs) for a stratified simple random sample of 3000 CT slices from our CT scan testing set without any data augmentation (i.e. flips, rotations, etc.) because we wanted to generalize our Grad-CAMs to a standard view of Chest CT Scans. Heat-maps of the activation map from the CNN's last convolutional layer were created with a CT scan as input. This quantitative heat-map was then normalized to a range of [0, 1] and transformed into a visualization with a jet color scale from *Matplotlib* library in Python. Superimposing these colored heat-maps onto the original CT scan highlights regions of the CT scan that the model perceives as significant for accurate classification. The Grad-CAMS show that lesions are generally present in the center of the lungs in bacterial pneumonia. Lesions - specifically GGOs for COVID-19 typically present peripherally, closer to the pleura. COVID-19 lesions are also shown to be much more scattered while lesions from bacterial pneumonia are more localized. These human-interpretable image features can be used by radiologists to improve the accuracy of manual diagnosis.

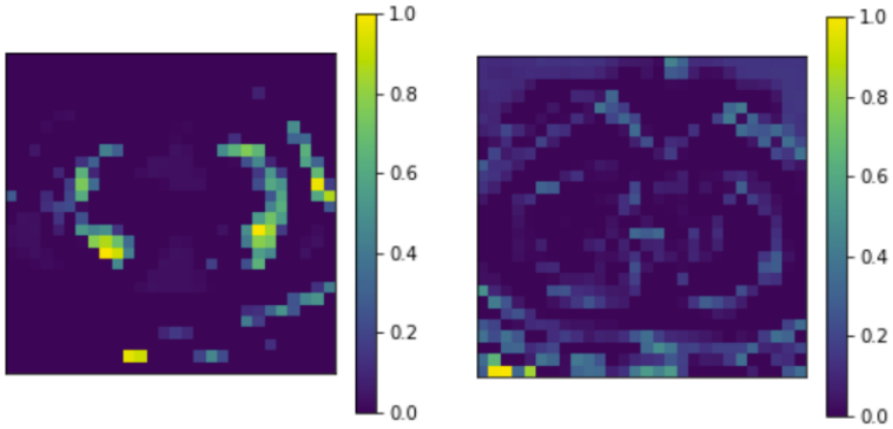

Figure 11. Grad-CAMs for bacterial pneumonia (left), and a COVID-19 CT scans (right). Yellow pixels have a high weightage. Blue pixels have a low weightage. Source: [Authors]

3.5 CFNN Testing

The clinical factors neural network (CFNN) was blindly tested on the remaining 7,200 sets. The CFNN achieved an accuracy of 88.75%, correctly classifying 6390 sets of clinical factors. The highest categorical accuracy of 97.58% came from the Healthy class, correctly predicting 2342 of the 2400 sets. The correct predictions out of the total 2400 sets for the pneumonia and COVID-19 classes were 1997 and 2051 sets respectively. Therefore, the CFNN should be used in conjugation with other models to produce the most accurate diagnosis. We ensemble the CFNN with our novel CNN in Section 4.1

3.6 Clinical Factor Neural Network Weights

The weights from the trained CFNN were extracted from the instantiated model to determine the importance of each clinical factor. These weights for the neurons mathematically transform the input into the output for the neuron and determine the impact of the neuron on the next layer. Using the *get_weights* function from the *layers* module in *tensorflow.keras*, the weights across the first layer were averaged for each of the 7 input neurons. After normalizing the weights to a [0, 1] range, we found that the most influential factor was “Shortness of Breath”. Weights closer to 0 have a smaller impact on the model’s prediction while weights closer to 1 have a larger impact on the prediction..

Clinical Factor:	Gender	Age	Cough Details	Shortness of Breath	Headache	Sore throat	Fever
Weight	0.424	0.243	0.389	1.000	0.694	0.111	0.622

Table 4. Normalized weights for each clinical factor from trained CFNN. Source: [Authors]

4. WEBSITE APPLICATION

4.1 Ensemble Model

To combine trained CNN and CFNN into our website, an ensemble model was created with the two networks. The predictions of each network are combined through weighted averaging. The calculations for these weights are based off of Federated Averaging from Federated Learning,[20] which determines a weight (w) based off of the ratio of training data used for the K^{th} model (n_k) to total training data used for all models (n).

$$w = \sum_{k=1}^{K} \frac{n_k}{n} * F(k)$$

For our ensemble model, K = 2 for the two trained models and F(k) are the weights for the k^{th} trained model. n_{CNN} = 105000 for the CNN and n_{CFNN} = 36000 for the CFNN which forms a ratio of 0.745 to 0.255 between the two models for the weighted average.

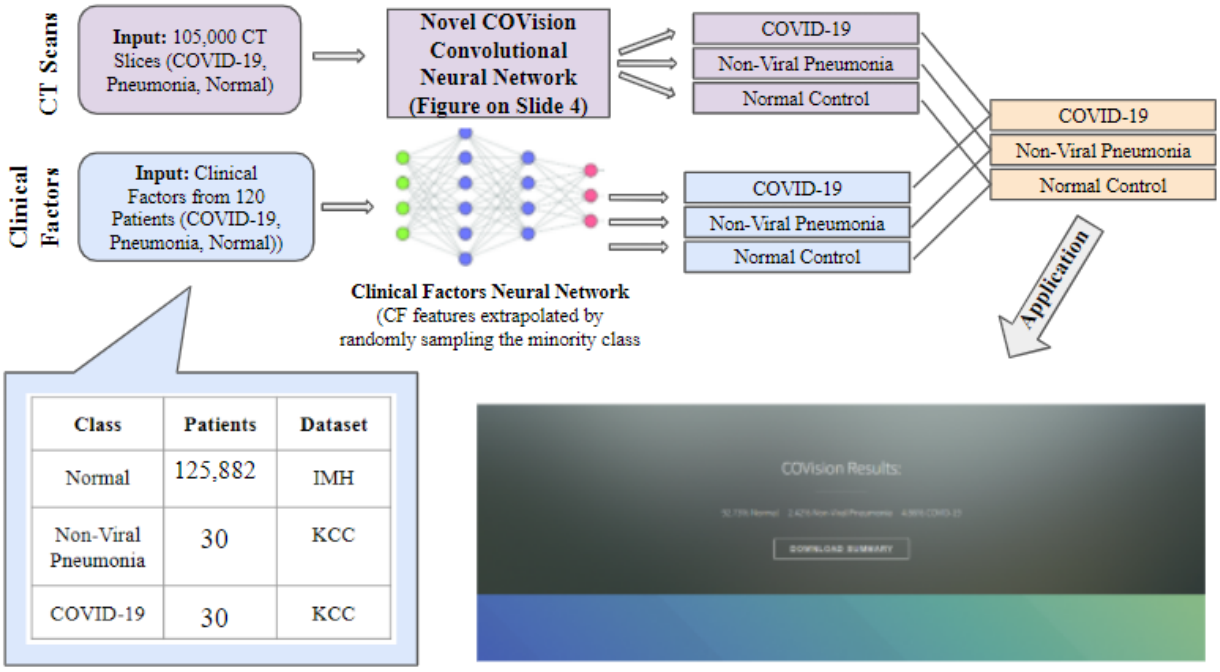

Figure 12. Overview of ensemble model combining our CNN and CFNN. Source: [Authors]

4.2 Website Deployment

A public website was created to deploy our ensembled model. The framework and styling of the website was developed with HTML and CSS with a Visual Studio Code text

editor. After importing the Tensorflow.js library as a script file, the ensemble model is integrated into the website by loading a JSON file with the trained weights. With the model loaded, the user can upload an isolated CT Slice or a CT volume for a patient and then make a selection of the patient's clinical factors. Once the data is inputted, the user can make a prediction through the ensemble model which outputs the probabilities of the patient having COVID-19, pneumonia, or is healthy. A summary of the diagnosis along with the clinical factors can be downloaded as a .txt file making results easily shareable.

Figure 13. COVision's web application diagnostic framework. A user would enter a CT slice/volume and/or clinical factors. Results are displayed as percentages for COVID-19, pneumonia, or healthy and can be downloaded as a .txt file. Source: [Authors]

4.3 Physician Feedback

Dr. Robin Thomas M.D. is a board-certified family medicine specialist with 10 years of experience.

"It is true that radiologists aren't able to differentiate consistently between bacterial pneumonia and COVID-19 pneumonia, so your radiology study makes sense and is valid. Therefore, COVision is practical in radiology circles and or for academic purposes to differentiate covid pneumonia from other viral pneumonia. If trained with more data, COVision definitely has the potential to be used in hospitals. COVision can even be used for identifying new types of viruses if it recognizes that a CT scan does not meet any of the other current viruses. I feel like the biggest highlight of your model for the medical community is that it can easily be used to improve upon and has the potential to diagnose between the types of pneumonia that a PCR test can't detect [since PCR tests have a low sensitivity]."

5. DISCUSSION

Through our research, we developed a novel deep learning framework to differentiate COVID-19 from other common pulmonary conditions with a high specificity - 98% . This framework can be trained with other lung conditions apart from bacterial pneumonia such as lung cancer. COVision is extremely lightweight with only 8 layers, takes a fraction of the time to train than state-of-the-art architectures, and is able to prevent overfitting with its simple structure. COVision achieved an accuracy of 95.8% and an AUROC of 0.970 on 25658 testing CT slices. When compared to three board certified radiologists with at least 5 years of experience, COVision has a statistically significant higher accuracy (95.8% vs. 73.4%), especially in differentiating COVID-19 from pneumonia and healthy CT Scans. After analyzing COVision's activation maps, we found evidence that COVID-19 lesions presented peripherally, closer to the pleura while pneumonia lesions presented centrally on a chest CT scan of the lungs (coronal plane). When analyzing the weights of our CFNN (clinical factor neural network), shortness of breath was the best indicator of COVID-19. Lastly, COVision is implemented into a cloud-based web application that can provide immediate feedback on diagnosis and treatment using CT Scans and basic clinical factors. COVision has the potential to save countless lives, particularly in developing nations with a shortage of doctors and huge volume of patients due to the coronavirus pandemic.

6. COVISION WEBSITE

Three hypothetical sets of patient data are provided to test the model. Note that COVision takes 3 seconds to load (initialization). Website: **https://covision.timmy625.repl.co/**

7. DATA AVAILABILITY

CT Scans of COVID-19, pneumonia, and healthy patients were obtained from the China Consortium of Chest CT Image Investigation (CC-CCII) dataset: http://ncov-ai.big.ac.cn/download?lang=en. Ground truth for the CC-CCII dataset was established via serology tests and confirmed by laboratory findings. Clinical factors for COVID-19, and pneumonia patients were obtained from the Khorshid COVID Cohort (KCC) study: https://figshare.com/articles/dataset/COVID-19_and_non-COVID-19_pneumonia_Dataset/16682422. Clinical factors for healthy patients were obtained from Israeli Ministry of Health public dataset: https://data.gov.il/dataset/covid-19/resource/74216e15-f740-4709-adb7-a6fb0955a048. We compiled all the clinical factors data into a CSV file using the *pandas* and *numpy* libraries in Python. We removed the clinical factors from the dataset that were not one of the following: shortness of breath, cough, headache, fever, sore throat, age, and gender. We binarized the ages of the patients by assigning a threshold age of 60 years.

8. CODE AVAILABILITY

All machine learning code was written in Google Colab using Python (v3.6.0). To design our novel architectures, we used *tensorflow* library (v2.8.0). All figures were plotted using the *matplotlib* library in Python. All website code was written in Visual Studio Code using HTML, CSS, and JavaScript. GitHub Repository: https://github.com/Kushy0814/COVision

9. REFERENCES

[1] *WHO Coronavirus (COVID-19) Dashboard*. (n.d.). With Vaccination Data.

https://covid19.who.int/

[2] *COVID-19 Lung Damage*. (n.d.). Johns Hopkins Medicine.

https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/what-coronavirus-does-to-the-lungs

[3]Kortela, E., Kirjavainen, V., Ahava, M. J., Jokiranta, S. T., But, A., Lindahl, A., Jääskeläinen, A. E., Jääskeläinen, A. J., Järvinen, A., Jokela, P., Kallio-Kokko, H., Loginov, R., Mannonen, L., Ruotsalainen, E., Sironen, T., Vapalahti, O., Lappalainen, M., Kreivi, H. R., . . . Kekäläinen, E. (2021). Real-life clinical sensitivity of SARS-CoV-2 RT-PCR test in symptomatic patients. *PLOS ONE*, *16*(5), e0251661. https://doi.org/10.1371/journal.pone.0251661

[4]Larremore, D. B., Wilder, B., Lester, E., Shehata, S., Burke, J. M., Hay, J. A., Tambe, M., Mina, M. J., & Parker, R. (2021). Test sensitivity is secondary to frequency and turnaround time for COVID-19 screening. *Science Advances*, *7*(1). https://doi.org/10.1126/sciadv.abd5393

[5] Hani, C., Trieu, N. H., Saab, I., Dangeard, S., Bennani, S., Chassagnon, G., & Revel, M.-P. (2020). COVID-19 pneumonia: a review of typical CT findings and differential diagnosis. *Diagnostic and Interventional Imaging*. https://doi.org/10.1016/j.diii.2020.03.014

[6] Self, W. H., Courtney, D. M., McNaughton, C. D., Wunderink, R. G., & Kline, J. A. (2013). High discordance of chest x-ray and computed tomography for detection of pulmonary opacities in ED patients: implications for diagnosing pneumonia. *The American journal of emergency medicine*, *31*(2), 401–405. https://doi.org/10.1016/j.ajem.2012.08.041

[7] Alhalaseh, Y. N., Elshabrawy, H. A., Erashdi, M., Shahait, M., Abu-Humdan, A. M., & Al-Hussaini, M. (2021). Allocation of the "Already" Limited Medical Resources Amid the COVID-19 Pandemic, an Iterative Ethical Encounter Including Suggested Solutions From a Real Life Encounter. *Frontiers in Medicine*, *7*. https://doi.org/10.3389/fmed.2020.616277

[8] Asghar, M. S., Yasmin, F., Alvi, H., Shah, S., Malhotra, K., Farhan, S. A., Ali Naqvi, S. A., Yaseen, R., Anwar, S., & Rasheed, U. (2021). Assessing the Mental Impact and Burnout among Physicians during the COVID-19 Pandemic: A Developing Country Single-Center Experience. *The American journal of tropical medicine and hygiene*, *104*(6), 2185–2189. https://doi.org/10.4269/ajtmh.21-0141

[9] Lambin P, Rios-Velazquez E, Leijenaar R, Carvalho S, van Stiphout RG, Granton P, Zegers CM, Gillies R, Boellard R, Dekker A, Aerts HJ. Radiomics: extracting more information from medical images using advanced feature analysis. Eur J Cancer. 2012 Mar;48(4):441-6. doi: 10.1016/j.ejca.2011.11.036. Epub 2012 Jan 16. PMID: 22257792; PMCID: PMC4533986.

[10] Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M. Y. L., Zhu, J., Li, C., Hewett,

S., Dong, J., Ziyar, I., & Shi, A. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, *172*(5), 1122-1131.e9. https://doi.org/10.1016/j.cell.2018.02.010

[11] Kumar, A., Tripathi, A. R., Satapathy, S. C., & Zhang, Y. D. (2022). SARS-Net: COVID-19 detection from chest x-rays by combining graph convolutional network and convolutional neural network. *Pattern recognition*, *122*, 108255. https://doi.org/10.1016/j.patcog.2021.108255

[12] He, K. (2015, December 10). *Deep Residual Learning for Image Recognition*. arXiv.Org. https://arxiv.org/abs/1512.03385

[13] Simonyan, K. (2014, September 4). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv.Org. https://arxiv.org/abs/1409.1556

[14] Szegedy, C. (2015, December 2). *Rethinking the Inception Architecture for Computer Vision*. arXiv.Org. https://arxiv.org/abs/1512.00567

[15] *2019 Novel Coronavirus Resource*. (n.d.). Ncov-Ai.big.ac.cn. Retrieved March 2, 2022, from http://ncov-ai.big.ac.cn/download?lang=en

[16] Zhang, Z. (2018, May 20). *Generalized Cross Entropy Loss for Training Deep Neural Networks. . .* arXiv.Org. https://arxiv.org/abs/1805.07836

[17] Kingma, D. P. (2014, December 22). *Adam: A Method for Stochastic Optimization*. arXiv.Org. https://arxiv.org/abs/1412.6980

[18] COVID-19 and non-COVID-19 pneumonia Dataset. (2021). *Figshare.com*. https://doi.org/10.6084/m9.figshare.16682422.v1

[19] (2022). Data.gov.il. https://data.gov.il/dataset/covid-19/resource/74216e15-f740-4709-adb7-a6fb0955a048

[20] McMahan, B. H. (2016, February 17). *Communication-Efficient Learning of Deep Networks from Decentralized Data*. arXiv.Org. https://arxiv.org/abs/1602.05