COVID-19 Infections Image Classification Using Simplified Convolutional Neural Network

Kanika Gandhi
Electrical and Computer Engineering
Western University
London, Canada
kgandh27@uwo.ca

Karanpartap Singh Aulakh
Electrical and Computer Engineering
Western University
London, Canada
kaulakh4@uwo.ca

Kiranjot Kaur
Electrical and Computer Engineering
Western University
London, Canada
kkaur89@uwo.ca

Shivank Mittal
Civil and Environmental Engineering
Western University
London, Canada
smitta7@uwo.ca

Omar Awadallah
Electrical and Computer Engineering
Western University
London, Canada
oawadall@uwo.ca

***Abstract*—Viruses and viral infections have been a major problem in human life for centuries and different types of viruses have plagued many civilizations across history. In present days, COVID-19 has been the largest pandemic experienced, and mitigating the risk of virus transmissions is difficult due to population growth and urbanization which means more people are living in closer proximity. The large number of people living in one area with proximity makes it hard to monitor people and infections, also mass transportation across the globe via airplanes increases the rate of transmissions around the world. To mitigate this risk, image processing and classification using X-rays can be implemented. This paper examines the use of the AlexNet and simplified Convolutional Neural Network (CNN) architectures to predict COVID-19 inspections from X-ray images of the lungs, this helps to provide instant results and increases the efficiency of monitoring viruses. The AlexNet architecture resulted in the model overfitting with only 52% accuracy on unseen data, hence the simplified CNN architecture with few parameters is used to prevent overfitting on the small dataset of 503 images that is used in this paper. Hyperparameter tuning is implemented to find the best set of parameters to train the simplified CNN model which achieved an accuracy of 92% on unseen data.**

***Keywords*—Classification, Convolutional Neural Network, COVID-19, Chest X-Ray, AlexNet**

I. Introduction

A. Overview

The deadly COVID-19, caused by the Coronavirus SARS-CoV-2, first appeared in Wuhan, China, in 2019 and spread worldwide, resulting in a pandemic as declared by WHO. As per WHO statistics data, due to COVID-19, nearly 6.5 million people have died so far among 640 million confirmed cases. Coronavirus, or COVID-19, causes acute respiratory syndrome and has high transmissibility. The exchange of coughs, sneezes, and breath droplets to the infected person leads to the spread of the disease among the population, making COVID-19 a highly transmissible and infectious disease. COVID-19 was considered a pandemic because of its contagious nature and the number of infected people. The virus was infectious, and factors such as the proximity of people living close to each other, urbanization, globalization, free trade, and travel throughout the globe contributed to the outspread of the virus. Thus, a need to control the spread of the deadly disease among the population is necessary by testing quickly and effectively. COVID-19 is responsible for lung diseases such as pneumonia and acute respiratory distress syndrome (ARDS sepsis) airway Bronchitis which makes lung x-rays an excellent input for detecting COVID in person under observation using AI systems. In this paper, X-ray chest images are used to predict whether a COVID-19 infection exists or not to control the spread of the virus in a short time. AlexNet and simplified CNN architectures are used to achieve the best performance on the dataset used in this problem. The CNN will classify images into positive and negative cases, Such models can be used to develop AI systems that can make fast predictions based on images and ensure that stringent measures are implemented to decrease the spread of the Coronavirus.

B. Dataset Description

In this project, two repositories were used for the dataset's development to increase the number of X-ray images [1], [2].The first dataset is from figshare.com and the second one is from Kaggle.

The first dataset contains 900 images of chest X-rays, including a metadata CSV file for 453 images that has various attributes such as filename, patient_id, sex, age, view, label, pcr_test, survival, location, admission, symptom offset, also_had_ct, has_fever, has_cough, has_dyspnea, has_diarrhea, spo2, other_symptoms, medical_background, opacification, other, is_bilateral and URL, whereas the second dataset has two folders, i.e., Covid and Non-Covid consisting of 111 and 234 images respectively. During the data preprocessing stage, an equal number of positive and negative images were randomly selected as the original dataset contained more positive cases than negative ones to avoid bias. Moreover, reformat the images to jpg, and develop an excel file containing the images' file names and labels.

II. Background

Deep Learning uses algorithms similar to the human brain, where each node represents a brain neuron in the neural network to mimic its nature and functionality. Convolutional Neural Network is one such example of a deep neural network which profoundly used for image classification and object detection. The primary aim of CNN is to compress the images strategically to prevent losing key features while processing large datasets. This process is performed by convolutional, max pool, stride, and fully connected layers.

This article compares the proposed CNN architecture with the existing AlexNet. The AlexNet has five convolutional layers, three max pool layers, and three fully connected dense layers; however, the proposed architecture has only two convolutional layers, one max pool layer, one flatten, and two fully connected dense layers, as shown in the Figure 1. The architecture of AlexNet is modified according to the problem and to reduce the computational complexities in the algorithm. Where AlexNet has around 61 million parameters, the simplified CNN has about 10,000 parameters which reduces the variance problem and help in preventing overfitting.

For image classification problems, several evaluation metrics are used to assess the accuracy of the model. In this paper, accuracy score, f1-score, precision, and recall are the metrics that are used to evaluate the model. Accuracy is the number of correct predictions over the whole sample, f1-score is another measure of the model's accuracy on the dataset. Precision is the value that shows how many of the predictions labeled as positive belong to the positive class. The recall represents the number of predictions correctly labeled as positive out of all the positives in the data. Equations 1,2,3, and 4 show the equations for accuracy, f1-score, precision, and recall respectively.

$$accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

$$F1\_score = 2 * \frac{(precision * recall)}{(precision + recall)} \tag{2}$$

$$precision = \frac{(TP)}{(TP + FP)} \tag{3}$$

$$recall = \frac{(TP)}{(TP + FN)} \tag{4}$$

III. Related Work

Coronavirus-2019 (COVID-19) caused severe health crises worldwide after its first appearance in Wuhan, China; its high transmissibility caused respiratory syndrome, making it a deadly outbreak that mankind has ever seen. Many researchers have done many studies to utilize deep learning and machine learning techniques to quickly detect the virus in radiological images in a cost-effective manner [3]. Image processing is the art of manipulating or using images to retrieve or draw conclusions, output, or decisions based on the image's analysis. Image processing is primarily used in the medical industry and has been used intensively since the unfortunate pandemic for detecting COVID-19 cases.

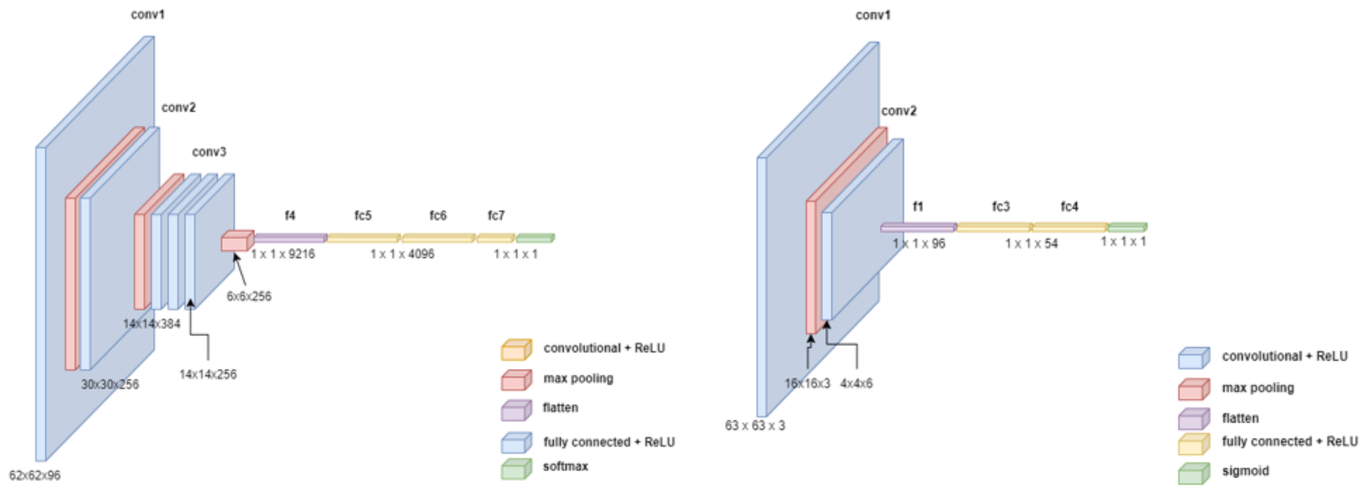
Multiple studies have been conducted in the past few years debating over which methodology can improvise the architecture of the implemented algorithm or technology to aid the detection of COVID-19 within individuals. In recent studies, it has been observed that by using artificial intelligence-based image processing techniques, the detection of abnormalities in chest C.T. scans could be an effective way to distinguish COVID-19-positive cases via image segmentation.. The extension of the investigation went to the extent of using the ResNet deep network to separate the COVID-19 cases from the pneumonia-ridden cases. Kaheel et al. (2021) [4] proposed an architecture efficient in detecting even minimalistic severity in COVID-19 cases, effectively allowing instant quarantine in positive patients.

A significant modification was brought into light when convolution neural networks were used amply to detect COVID-19. One such novel modification stole the light when the evolution of stacked CNN came forward to perform the segregation of normal, COVID-19 and pneumonia X-ray images. Gour et al. (2021) [5] developed an automatic COVID-19 diagnostic system using chest X-ray and C.T. images by implementing a stacked CNN model. This approach obtained different pre-trained CNN sub-models from VGG19 and Xception during the training. This model learned the discriminative features and then implied the learning to identify the diverse features in the chest X-rays to distinguish, resulting in higher accuracy with pre-trained CNN sub-models over existing models.

The world experienced an exponential increase in COVID-19 cases, resulting in the failure of the medical infrastructure, causing inaccurate diagnosis and treatment of the disease. Therefore, Ayalew et al. (2022) [6] proposed a quick diagnostic approach by combining a hybrid CNN and histogram of oriented gradients (HOG), yielding a highly accurate detection and classification technique for COVID-19 using X-ray images. The proposed hybrid architecture model greatly reduced the chances of false positives and negatives. The proposed methodology was used for feature extraction, whereas YOLOv3 and SVM were used for object detection and classification.

Due to the SARS-CoV-2 virus, severe respiratory syndrome was caused, resulting in major health crises throughout the world. Zouch et al.(2022) [7] performed two different CNN models, the VGG19 and the ResNet50, to improve the automatic detection technique's performance using tomographic and radiographic images. The data augmentation was implemented (using random rotation, translation, and horizontal flip) due to the lack of a resourceful dataset to increase model accuracy. The model with data augmentation produced higher accuracy in radiographic images over no data augmentation.

The CT scan and X-Ray images are rich in patterns and



(a) (b)

Fig. 1: Architecture for (a) AlexNet and (b) simplified CNN

clusters to detect and confirm coronavirus presence in the lungs. Kumar et al. (2022) [8] presented a comprehensive analysis of the detection of COVID-19 using a deep-learning technique as it could be deemed reliable and affordable. This paper studied the cost-effectiveness of the proposed method with other existing methodologies, and several -financial-related aspects were also discussed from the perspective of insurance companies.

The foundation of transfer learning methodology followed the modification and exploitation of preceding models to save the time it takes to train a new one from scratch. Gupta, Vedika, et al. (2022) [9] aimed to distinguish healthy patients from those suffering from pneumonia and COVID-19-positive using X-ray images by implementing pre-trained AlexNet, VGG16, ResNet, and MobileNetV2. The precision, recall, and F1-score were compared for all the models, which concluded that AlexNet was outstanding among the others. Yang et al. (2021) [10] carried out a similar study on a larger dataset using the VGG16, DenseNet121, ResNet50, and ResNet152 models. The research was carried out on the same grounds of transfer learning logic; it yielded somewhat different results due to various comparative factors. In comparison to the others, VGG16 showed better accuracy results in this scenario. The main output of such analysis can act as a bedrock to help carry out more ingenious studies specific to the pandemic that will outshine the previous Computer-aided design systems.

In comparison with the related work, the model this paper proposes is a modification of previously implemented CNN providing a debatable comparison with the pre-existing AlexNet in a novel manner. The existing conventional architecture of CNN is specific to the vast horizon of datasets, varying largely in size. The simplified CNN model is developed to curb problems that could arise if the existing complex models are applied to the small dataset used in this paper. The basis of tempering and assembling a simpler version of CNN is to prevent the model from learning data infused with outliers or noise, mitigate overfitting, and reduce the computational cost and training time of the model.

IV. Methodology

A. Data Preprocessing

As mentioned in the introduction section, the first dataset [1] that is used for this problem contains a total of 900 X-ray images of the lungs with information for 452 images only; 359 images represent positive COVID-19 cases, 23 images represent negative COVID-19 cases, and 70 unlabeled images. Therefore, the dataset is imbalanced as it has more positive samples with information than negative samples which increases bias. Before balancing the data, the first dataset is cleaned by dropping all columns except the image filename and the label *pcr_test* columns. All unlabeled images are dropped and only labeled images are used in the dataset. The final data for the first dataset after cleaning is shown in Figure 2, where there are two columns and a total of 382 samples (452 images – 70 unlabeled).

filename	pcr_test
260.jpg	positive
261.jpg	positive
262.jpg	positive
263.jpg	positive
264.jpg	positive

Fig. 2: First five rows of first dataset

A second dataset [2] is used to add more negative samples to the data to balance it. The second dataset does not have a labeled file format, instead, the images are split into two different folders, one for positive samples (Covid) and the other for negative samples (nonCovid). A CSV file is created using python with a filename and *pcr_test* label attributes, the code iterates over each image in both the Covid and nonCovid folders and populates the CSV file with each image's name and label. The final data for the second dataset after cleaning is shown in Figure 3, where there are two columns and a total of 341 samples (111 positives + 230 negatives).

	filename	pcr_test
0	PosCov0.jpg	Covid
1	PosCov1.jpg	Covid
2	PosCov10.jpg	Covid
3	PosCov100.jpg	Covid
4	PosCov101.jpg	Covid

Fig. 3: First five rows of second dataset

After the cleaning process is done, both datasets are combined which results in 470 positive samples and 253 negative samples for a total of 723 samples. The positive samples in the combined dataset are reduced to 250 samples by filtering out 220 positive samples, the negative samples are kept at 253 samples for a total of 503 samples. After balancing the data, one-hot encoding is used for the labels, positive COVID-19 cases are labeled as one and negative COVID-19 cases are labeled as zero. The final combined data is shown in Figure 4, with two attributes representing image file names and labels, and a total of 503 samples.

	filename	pcr_test
0	PosCov0.jpg	1
1	PosCov1.jpg	1
2	PosCov10.jpg	1
3	PosCov100.jpg	1
4	PosCov101.jpg	1
...	...	...
498	844.jpg	0
499	845-.jpg	0
500	882.jpg	0
501	883.jpg	0
502	888.jpg	0

Fig. 4: Combined dataset with numeric labels

The next step is feature generation, where the file names are used to obtain the pixels for each image. The file names are used to iterate over each image in the dataset to resize all images to 256x256 pixels to keep all images of consistent sizes, all images are also converted to grayscale to reduce the complexity from three layers to one layer which decreases the computational cost and time to train the model. Once images are converted into grayscale, the code iterates over each image and obtains the pixel values, these pixel values are flattened and populated to a data frame X that represents the inputs for the CNN model. Since the image sizes are 256x256 pixels, the data frame contains 256x256 = 65536 columns and 503 rows, where each row represents one image, and the columns represent the pixel values for that specific image. The input data frame is shown in Figure 5.

	0	1	2	3	4	5	6	7	8	9	...	65526	65527	65528	65529	65530	65531	65532	65533	65534	65535
0	22.0	21.0	20.0	19.0	19.0	20.0	20.0	20.0	24.0	24.0	...	22.0	22.0	21.0	22.0	21.0	19.0	19.0	20.0	18.0	15.0
1	43.0	46.0	49.0	52.0	57.0	50.0	59.0	49.0	52.0	54.0	...	81.0	78.0	78.0	76.0	71.0	68.0	68.0	65.0	79.0	65.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	121.0	116.0	106.0	95.0	86.0	84.0	70.0	55.0	46.0	31.0
3	22.0	23.0	26.0	26.0	27.0	22.0	40.0	46.0	51.0	51.0	...	181.0	163.0	169.0	152.0	138.0	132.0	114.0	93.0	89.0	90.0
4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	1.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Fig. 5: Pixels input dataframe of size 256x256

The input data frame X is normalized by dividing all pixels by 255 which converts the pixel range from [0-255] to [0-1], this normalization ensures similar data distribution and leads to faster convergence when training the CNN model. Finally, an output data frame y is created using the numeric image labels shown in Figure 4.

B. Model Training

For the training and validation process, hold-out validation is implemented by splitting the data into training, validation, and testing sets with 70% (352 samples), 20% (101 samples), and 10% (50 samples) respectively. The training and validation sets are used in the tuning process to validate each set of parameters to choose the best set of parameters for the model on this dataset. To feed the images into the CNN model, the inputs X_train and X_val are reshaped to the (height, width, channel) format as (256, 256, 1), and the height and width are set to 256 because each image was resized to 256x256 pixels in the preprocessing part, the channel has a value of one because all images are grayscale images which have only one channel as opposed to colored RGB images that have three channels.

The hyperparameter tuning process is performed on the CNN model described in the background section, this CNN model consists of two convolutional layers, a max-pooling layer between them to reduce the dimensions of feature maps and thus reduce parameters, three fully connected layers, and an output layer with a sigmoid activation function. Each layer contains a kernel L2 regularization, the L2 regularization is used to reduce over-fitting in the model. A dropout layer is also added between the first and second fully connected layers to reduce overfitting. Finally, the output layer uses the sigmoid activation function because the dataset only has two classes. The sigmoid function produces an output between 0 and 1 with a threshold of 0.5 which means if the output is greater than 0.5 it will represent label 1 or positive samples, otherwise, it will represent label 0 or negative samples.

Once the CNN model is built, the training and validation sets are used to tune the model using hyperband tuning with early stopping. Hyperband tuning is faster than normal tuning techniques such as grid search. Hyperband tuning saves time by performing random sampling on parameter sets and running sets for a few iterations to understand how they perform, then it saves the iterations with the best performance and trains them for a longer time, and ignores configurations with poor performance. In short, hyperband uses earlier results to select the best performers to train them longer, if the performance decreases or is constant, the training stops early, which reduces the total time it takes to perform hyperparameter tuning.

C. Evaluation Procedure

To evaluate both the original AlexNet model and the updated CNN model, plots of the accuracy measures for the training process are generated and compared. The trained model is evaluated on 50 testing samples to produce a total of 50 predictions for COVID-19 cases, the predictions are used to generate confusion matrices for both models and these matrices are compared. The information from the confusion matrix is used to calculate the precision, recall, accuracy, and f1-score metrics for the model, these metrics are compared for each model. Finally, the receiver operating characteristic (ROC) curve is generated and the area under the curve (AUC) is computed, both the ROC curve and AUC value are compared for both models.

V. Results

Before training the CNN model, hyperparameter tuning is done by tuning the number of neurons in the input convolutional layer, the L2 regularization value for each layer, the drop rate of the dropout layer, and the learning rate of the Adamax optimization function. The number of neurons is an important hyperparameter that affects the network operation, the L2 regularization and dropout rate values affect how much overfitting is prevented, and the learning rate is responsible for how fast the weights are updated which affects the network operation, therefore all four parameters are tuned. Table I summarizes the important hyperparameters that affect the CNN model, the tuned parameter ranges, and the best set of values.

TABLE I: Summary of tuned hyperparameters

Parameter	**Description**	**Range**	**Best parameters**
Filters	Number of neurons in the input layer	1 to 10	3
Kernel_regularizer	L2 regularization value	0.0001, 0.001, 0.01,0.1,1	0.001
Learning rate	Step size to update weights	0.000001, 0.00001, 0.0001, 0.001, 0.01 ,0.1,1	0.001
Dropout rate	Rate to drop noisy data	0.1,0.2 ,0.3, 0.4,0.5	0.4

The best parameters selected are used to train the tuned CNN model while the original AlexNet model is trained using the default parameters. Using early stopping, the AlexNet model training process stopped at 50 epochs and the tuned CNN model is trained for 200 epochs. The training and validation accuracy measures of both models are shown in Figure 6.

From the accuracy measures plots, the initial model, the AlexNet model, that was used on the dataset showed poor performance in the training process. It is observed that the

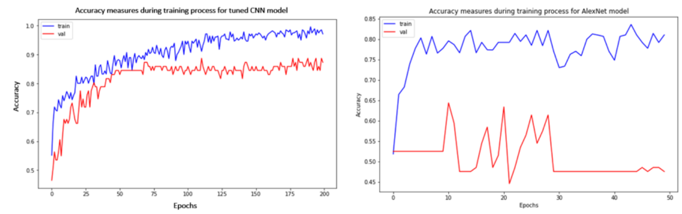
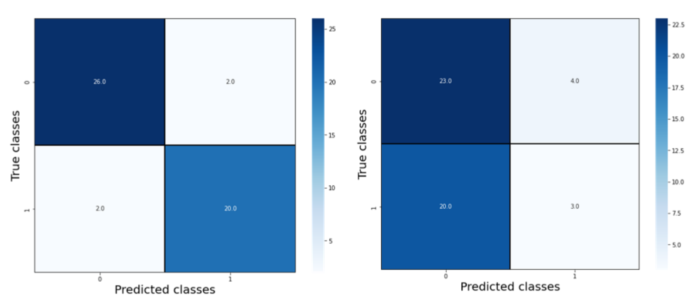


Fig. 6: Accuracy plots for CNN and AlexNet architectures

training accuracy has an increasing trend, and the model is successfully learning on the training set, however, the model performance is very poor on the validation set and the model does not generalize well on the unseen data. The training accuracy increases from around 52.5% to around 80%, while the validation accuracy decreased from 52.5% to around 47.5% and is not learning. This behavior indicates the presence of overfitting because the model performs well on the training data while it performs poorly on unseen data. The AlexNet model has around 60 million parameters in total, while the dataset only consists of 503 samples, the huge number of parameters that are used on a very small dataset leads the model to memorize noises that fit too closely to the training data and result in overfitting. Therefore, the CNN model is updated to reduce the complexity of the architecture to reduce the number of parameters. The tuned CNN model is much smaller than the original AlexNet model, it only includes about 10,000 parameters in total which is significantly lower than 60 million parameters. When training the simplified CNN model, it is observed that the tuned CNN model is successfully learning because there is an increasing trend in the training and validation accuracies as training iterations increase. The training accuracy starts at around 55% and ends at around 99% while the validation accuracy starts at around 45% and ends at around 88%-90%. The training and validation curves successfully converge which also indicates that the CNN model is learning. After both models are trained, the models are used on the testing set of 50 images to predict whether the image represents a positive or negative COVID-19 case. The predicted values are used to generate the confusion matrices that are shown in Figure 7. The True-positive (TP), True-negatives (TN), False-positives (FP), and False-negatives (FN) values from the confusion matrices are used to compute the evaluation metrics shown in Table II.



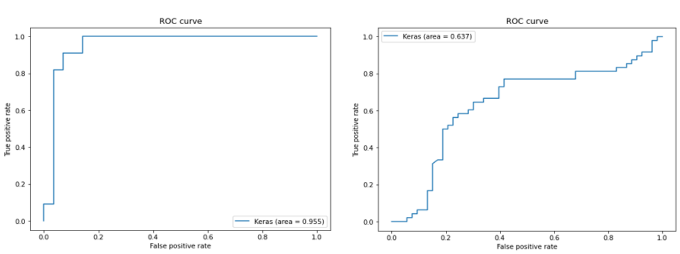
(a) (b)

Fig. 7: Confusion matrices for (a) tuned CNN model and (b) AlexNet model

TABLE II: Evaluation metric values for CNN and AlexNet models

Model	*Accuracy*	*F1-score*	*Recall*	*Precision*
Tuned CNN Model	0.92	0.9091	0.9091	0.9091
AlexNet Model	0.52	0.2	0.1304	0.4286

The confusion matrices show that the tuned CNN model performs better on unseen data as opposed to the complex AlexNet model. CNN model predicted 26 negative samples and 20 positive samples correctly and predicted only four samples incorrectly out of 50 samples. The TN, TP, FN, and FP values for the CNN model are 26, 20, 2, and 2 respectively. AlexNet model predicted 23 negative samples and 3 positive samples correctly and predicted 24 samples incorrectly out of 50. The TN, TP, FN, and FP values for the AlexNet model are 23, 3, 20, and 4 respectively. The evaluation metrics show that the CNN model is better overall than the AlexNet model. CNN model has equal F1, recall, and precision values because it predicted an equal number of samples, two, incorrectly for each class, the accuracy score is 92% which is 40% higher than the AlexNet model. The metrics for the AlexNet model show that the quality of positive and negative predictions is worse than the CNN model. The bad performance on unseen testing data supports the overfitting behavior that is observed earlier. Finally, the ROC curves for both models are shown in Figure 8. The ROC curve for the tuned CNN model converges more to the top-left corner of the plot than the AlexNet model. Also, the AUC value for the CNN model is very close to 1 which is very high and shows that the CNN model is capable of distinguishing between positive and negative COVID-19 cases. The AlexNet ROC curve does not converge to the top-left corner as well as the CNN curve, and the AUC value is also lower, which shows that the AlexNet model cannot accurately distinguish between positive and negative COVID-19 cases.



(a) (b)

Fig. 8: ROC curves for (a) CNN and (b) AlexNet models

VI. Conclusion

In this paper, a framework for image classification to predict COVID-19 infections from X-ray images of the lungs has been proposed. The classification was done using image processing, where images were processed using a simplified CNN architecture. This paper initially used the AlexNet architecture and then used a simplified CNN architecture with an early stopping technique to classify whether a patient has a positive or negative COVID-19 result from the X-ray image of their lungs. The simplified architecture was implemented to prevent overfitting and decrease computational cost, and the early stopping technique was used to speed up the training process. From the results, it is concluded that the AlexNet architecture successfully learned on the training set, however, it achieved poor performance on unseen data with only 52% accuracy. The F1 score, precision, recall, and AUC values are all below 0.7 which shows that the AlexNet architecture is not able to distinguish between positive and negative COVID-19 cases. The bad performance of the AlexNet is due to the huge number of parameters, around 60 million, that are used on a small dataset which led the model to overfit and perform poorly on unseen data as opposed to the training data. The simplified CNN architecture with reduced parameters, around 10,000, was used. It is concluded that the CNN model is better overall as it achieved an accuracy of 92% and an F1 score, precision, recall, and AUC values of above 0.9. The accuracy measures during training showed that the CNN model has successfully learned, and both the training and validation accuracy measures increased over the iterations and successfully converged. In conclusion, the simplified CNN architecture with fewer parameters is the best to train on the dataset used in this paper.

References

[1] Figshare. (2021), “COVID-19 Chest X-Ray Image Repository”. Website, 2021. https://figshare.com/articles/dataset/COVID-19_Chest_X-Ray_Image_Repository/12580328

[2] Kaggle. (2021), “COVID-19 Chest X-Ray Images”. Website, 2021. https://www.kaggle.com/datasets/mohammadamireshraghi/covid19-xray-local-data

[3] Costa, Y. M., Silva Jr, S. A., Teixeira, L. O., Pereira, R. M., Bertolini, D., Britto Jr, A. S., Cavalcanti, G. D. (2022). COVID-19 Detection on Chest X-ray and CT Scan: A Review of the Top-100 Most Cited Papers. Sensors, 22(19), 7303.

[4] H. Kaheel, A. Hussein, A. Chehab, “AI-Based Image Processing for COVID-19 Detection in Chest CT Scan Images,” Front. Comms. Net 2:645040, 2021.

[5] M. Gour, S. Jain, “Automated COVID-19 detection from X-ray and CT images with stacked ensemble convolutional neural network,” biocybernetics and biomedical engineering 42, pp. 27-41, 2021.

[6] A. Ayalew, A. Salau, B. Abeje, B. Enyew, “Detection and classification of COVID-19 disease from X-ray images using convolutional neural networks and histogram of oriented gradients,” Biomedical Signal Processing and Control 74, 103530, 2022.

[7] W. Zouch, D. Sagga, A. ECHTIOUI, R. KHEMAKHEM, M. GHORBEL, C. Mhiri, A. BEN HAMIDA, “Detection of COVID-19 from CT and Chest X-ray Images Using Deep Learning Models,” Annals of Biomedical Engineering, Vol. 50, No. 7, pp. 825–835, 2022. https://doi.org/10.1007/s10439-022-02958-5

[8] M. M.V., S. Atalla, N. Almuraqab, I. Moonesar, “Detection of COVID-19 Using Deep Learning Techniques and Cost Effectiveness Evaluation: A Survey,” Front. Artif. Intell. 5:912022, 2022.

[9] V. Gupta, N. Jain, J. Sachdeva, M. Gupta, S. Mohan, M. Bajuri, A. Ahmadian, “Improved COVID-19 detection with chest x-ray images using deep learning,” Multimedia Tools and Applications 81:37657–37680, 2022. https://doi.org/10.1007/s11042-022-13509-4

[10] D. Yang, C. Martinez, L. Visuña, H. Khandhar, C. Bhatt, J. Carretero, “Detection and analysis of COVID 19 in medical images using deep learning techniques,” Scientifc Reports 11:19638, 2021. https://doi.org/10.1038/s41598-021-99015-3