Deep Learning

880008-M-6

Assignment

Using Deep Learning to Perform Multi-Class Classification on the Covid19 Chest X-ray Dataset

Report by:

NIKOLAOS MITSAS (2090834)

Group Number:

10

Group Members:

ALEJANDRO RIVERA LOPEZ
BAS RONGEN

CHIARA MANNA

March 2023

1. Problem Definition

X-ray images are grayscale medical images commonly used for detecting various illnesses. For instance, COVID-19 and pneumonia can be detected by analyzing X-ray images of the lungs. In recent years, machine learning and deep learning models have been increasingly used in the medical domain to develop automatic diagnostic systems that can support medical professionals in detecting and treating respiratory illnesses. Among these models, convolutional neural networks (CNNs) have emerged as a powerful tool for medical image analysis, (Reshi, A. A. et al, 2021).

The objective of this research is to develop a diagnostic system for COVID-19, bacterial pneumonia, viral pneumonia, and healthy lungs using X-ray images. Three different models will be used and evaluated in this study. The first model is already built, it will use a standard CNN architecture and it will be used as a baseline. The second model will be an improvement of the first CNN model, aiming to outperform it. Finally, the last model will be based in transfer learning with VGG16. By comparing the performance of these models, this study aims to identify the most effective approach for detecting COVID-19 and different types of pneumonia using X-ray images.

2. Dataset Preprocessing

For the preprocessing of the data, the images were first resized to a fixed size of 156x156 to ensure that all images had the same size. Subsequently, the categorical labels were turned into integers using the label encoder function from scikit-learn library, since deep learning models require numerical inputs. The data were split and stratified into training set (60%), validation set (20%) and testing set (20%), using the split_train_test function from the same library, to avoid overfitting, evaluate the performance of the model and contact hyperparameter tuning. To prepare the labels for a multiclass classification task, to_categorical function from the Keras library was used to transform them into a binary class matrix. The data was then converted to numerical type and normalized so they would be adjusted in similar scales. Normalization was achieved by dividing the data with the highest possible value, 255.

After preprocessing the data, exploratory data analysis (EDA) was conducted to gain insights into the dataset and identify any potential issues that might impact the performance of the model. During EDA, class imbalance was detected in the dataset, where COVID-19 class had a relatively low number of samples. In addition, some of the images in the dataset were not lung X-rays but CT scans, but it was decided not to handle them, since the dataset was provided.

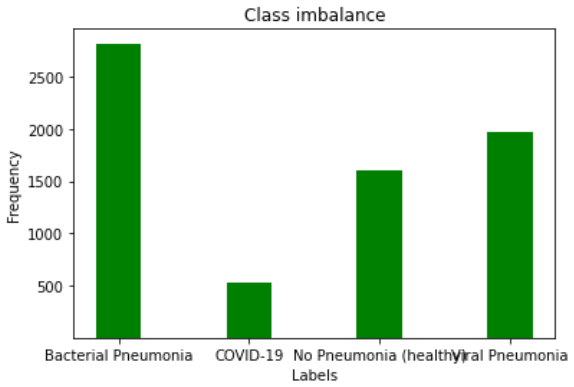

Figure 1: Class imbalance

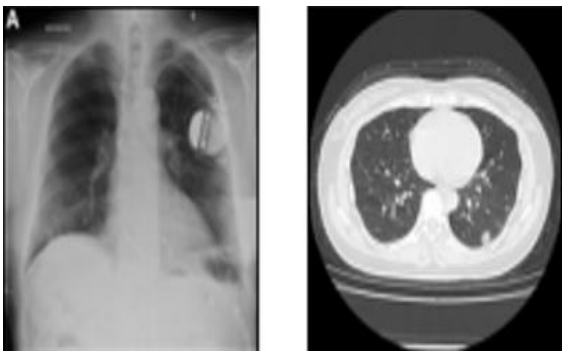

Figure 2: Visualized data

3. Baseline Model

The baseline model was provided. It constructed using 4 Conv2D layers with size (3, 3), where the first and third layers had 64 nodes, and the second and fourth layers had 32 nodes. All the layers were activated using the Rectified Linear Unit (ReLU) activation function, and padding was set to "same". Right after the second and fourth Conv2D layer, maxpooling layers were also added with a size equal to (2, 2). Following the Conv2D layers, a flatten layer was applied, followed by two dense layers with ReLU activation functions, having both 32 nodes. The output layer contained 4 units since the

problem was a multi-class classification task with 4 classes, and the activation function used was SoftMax. In conclusion, the model was compiled utilizing the Adam optimizer, with a training process of 10 epochs and a batch size of 32.

After training and evaluating the multi-class classification model on a validation set, it was observed that the training and validation loss were decreasing until a certain point (epoch 4) while after that validation loss started to increase. The accuracy for both sets had similar behavior, indicating that the model was overfitting.

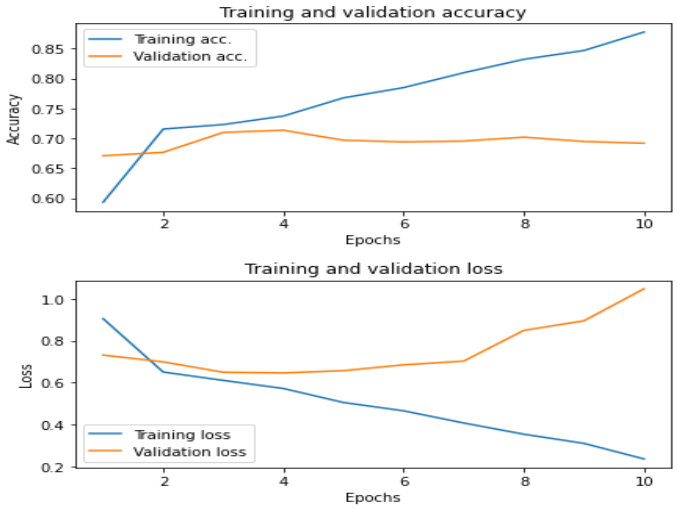

Figure 3: Baseline model train/validation loss and accuracy

That was the main problem of this model, since otherwise the micro average F1-score was not poor, around 68.3%, which suggests that it could identify most of the cases. For both validation and test confusion matrix it was obvious that the model had difficulties in identifying the minority class, COVID-19 and viral pneumonia.

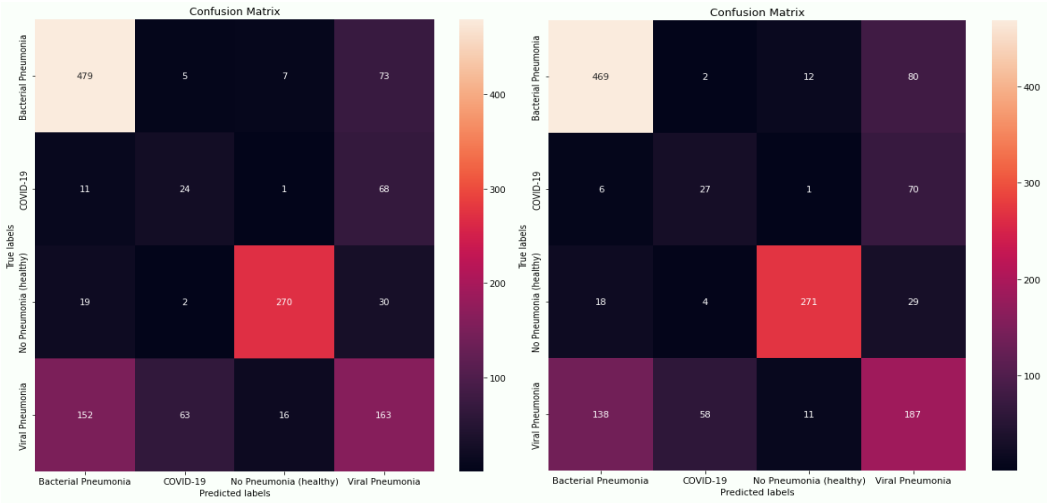

Figure 4: Validation confusion matrix for baseline model *Figure 5: Test confusion matrix for baseline model*

```
Class Bacterial Pneumonia:
Specificity :0.7778
Sensitivity: 0.8493
F1-score: 0.782
Accuracy: 0.8069

Class COVID-19:
Specificity :0.9453
Sensitivity: 0.2308
F1-score: 0.2424
Accuracy: 0.8915

Class No Pneumonia (healthy):
Specificity :0.9774
Sensitivity: 0.8411
F1-score: 0.878
Accuracy: 0.9458

Class Viral Pneumonia:
Specificity :0.8271
Sensitivity: 0.4137
F1-score: 0.4478
Accuracy: 0.7093


F1 micro average: 0.6829981718464352
```

Figure 6: Baseline Model metrics

This can also be seen for COVID-19 from the F1-score that was 24.24%, while test AUC score reaches 93%, figure 8. It was a result of true positive rate (sensitivity) that was also low, 23.08% and true negative ratio (specificity) at 94.53%. Viral pneumonia might have higher F1-score, 44.78% indicating that it performs better than COVID-19, but since specificity was higher and sensitivity lower, it did not seem to perform so well in the test ROC-AUC plot. Additionally, it was the closest curve to the change level curve. The test ROC-AUC plot for the 4 classes showed a slightly curved shape, with micro-average AUC score and macro-average AUC score being 92% and 89%, respectively. Finally, no pneumonia (healthy) seems to be classified better than all other classes for both F1-score (87.8%) and test AUC (98%). This was obvious from the test ROC-AUC plot, since no pneumonia was positioned closest to the top-left corner of the plot.

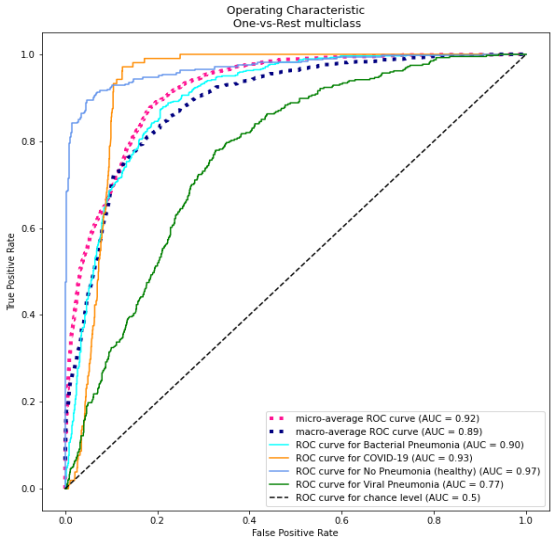

Figure 7: Validation ROC-AUC for baseline model

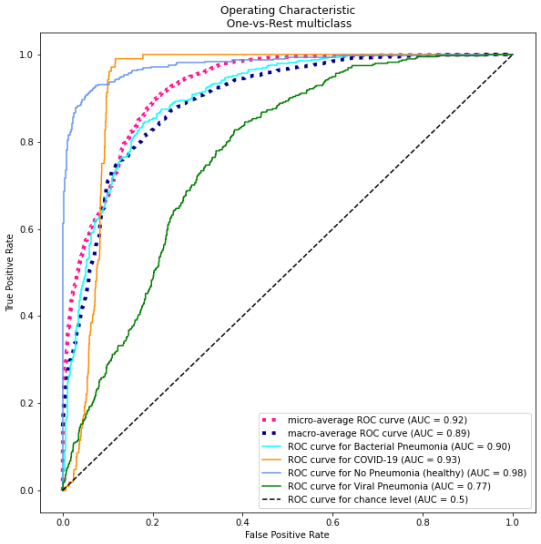

Figure 8: Test ROC-AUC for baseline model

4. Improved (Fine-tuned) Model

Prior to enhancing the baseline model, data augmentation was conducted on the training data with the aim of expanding the training set and mitigating overfitting, (Garcea, F. et al, 2022). After hyperparameter tuning the best result for data augmentation was with flipping the image horizontally and width and height shift at 0.1. Subsequently, the baseline model was improved, and overfitting was alleviated. To improve performance further, additional layers were introduced to the model, thereby increasing its complexity. The idea for more layers came from the model used in (Reshi, A. et al, 2021), which had promising results. However, this resulted in the dataset overfitting once more. To counter this, two new dropout layers were appended, which were similarly employed in the (Krizhevsky A. et al, 2017) reference. Consequently, the model exhibited improved performance, while successfully avoiding overfitting, as also seen in figure 9.

Throughout all the CNN layers, the padding value of 'same' was maintained, and for both conv2D and dense layer the ReLU function was utilized as the activation function. The model was subsequently recompiled and Adam optimizer was applied, due to its frequent usage in the field. After training the model and evaluating it over a period of 100 epochs, it was observed that the model performed optimally for40 epochs. The optimal batch size and number of nodes for each layer were chosen after experimentation with several different values.

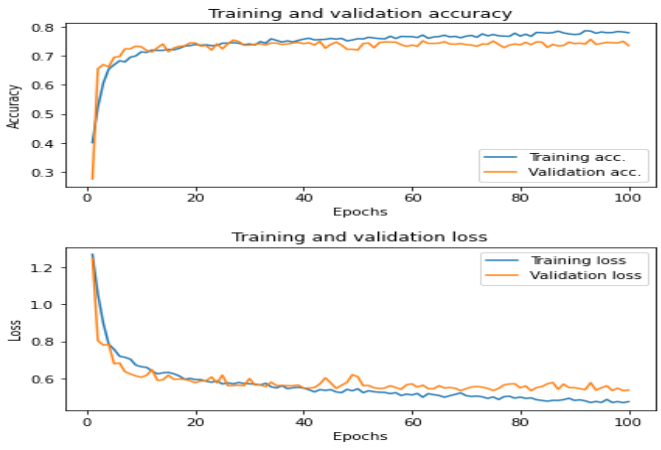

Figure 9: Improved model train/validation accuracy & loss

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv2d_18 (Conv2D)	(None, 156, 156, 64)	1792
conv2d_19 (Conv2D)	(None, 156, 156, 32)	18464
max_pooling2d_9 (MaxPooling 2D)	(None, 78, 78, 32)	0
conv2d_20 (Conv2D)	(None, 78, 78, 32)	9248
conv2d_21 (Conv2D)	(None, 78, 78, 32)	9248
max_pooling2d_10 (MaxPoolin g2D)	(None, 39, 39, 32)	0
conv2d_22 (Conv2D)	(None, 39, 39, 32)	9248
conv2d_23 (Conv2D)	(None, 39, 39, 16)	4624
max_pooling2d_11 (MaxPoolin g2D)	(None, 19, 19, 16)	0
conv2d_24 (Conv2D)	(None, 19, 19, 16)	2320
conv2d_25 (Conv2D)	(None, 19, 19, 16)	2320
max_pooling2d_12 (MaxPoolin g2D)	(None, 9, 9, 16)	0
flatten_3 (Flatten)	(None, 1296)	0
dense_9 (Dense)	(None, 64)	83008
dropout_3 (Dropout)	(None, 64)	0
dense_10 (Dense)	(None, 32)	2080
dense_11 (Dense)	(None, 4)	132

Total params: 142,484
Trainable params: 142,484
Non-trainable params: 0

Figure 10: CNN improve model summary.

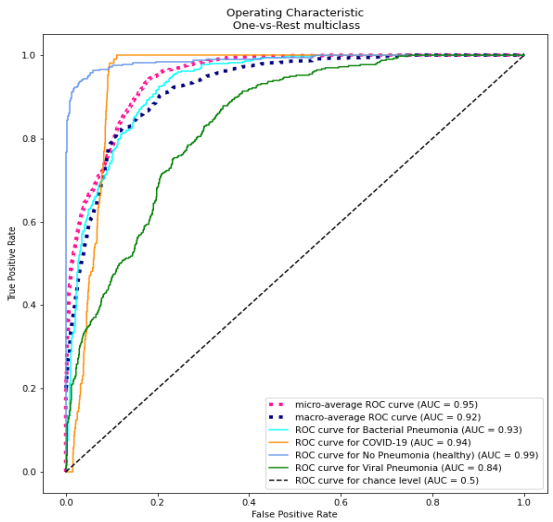

Figure 11: Validation ROC-AUC plot for improved model

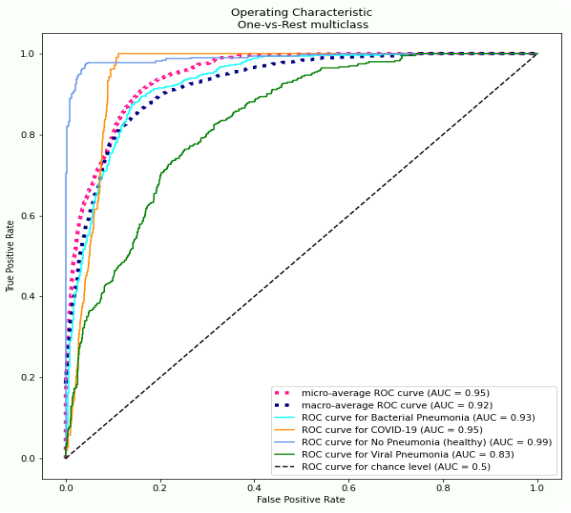

Figure 12: Test ROC-AUC plot for improved model

```
Class Bacterial Pneumonia:
Specificity :0.6728
Sensitivity: 0.8652
F1-score: 0.7394
Accuracy: 0.7513

Class COVID-19:
Specificity :0.9812
Sensitivity: 0.1538
F1-score: 0.2222
Accuracy: 0.919

Class No Pneumonia (healthy):
Specificity :0.9765
Sensitivity: 0.9315
F1-score: 0.9271
Accuracy: 0.966

Class Viral Pneumonia:
Specificity :0.907
Sensitivity: 0.434
F1-score: 0.5205
Accuracy: 0.7722

F1 micro average: 0.7322618580948648
```

Figure 13: Improved model metrics

Similar to the previous model, the AUC score for no pneumonia once again yielded the highest score at 99%, which could be attributed to the exceptional specificity and sensitivity of that class. Meanwhile, the AUC score for COVID-19 was also remarkably high, with a higher specificity by approximately 4% and lower sensitivity by 8%. Viral pneumonia, on the other hand, was positioned closer to the center of the ROC-AUC plot, indicating poorer overall performance. Nevertheless, the F1-score for viral pneumonia was greater than 50%, besting that of COVID-19 by approximately 30%, thus implying that the model exhibited greater precision in predicting cases of viral pneumonia.

Finally, it was clearly distinguishable from the confusion matrixes that bacterial pneumonia had the highest number of correctly classified instances. Conversely, while COVID-19 appeared to had the least number of correct classifications, but it should be noted that this was the imbalance minority class within the dataset.

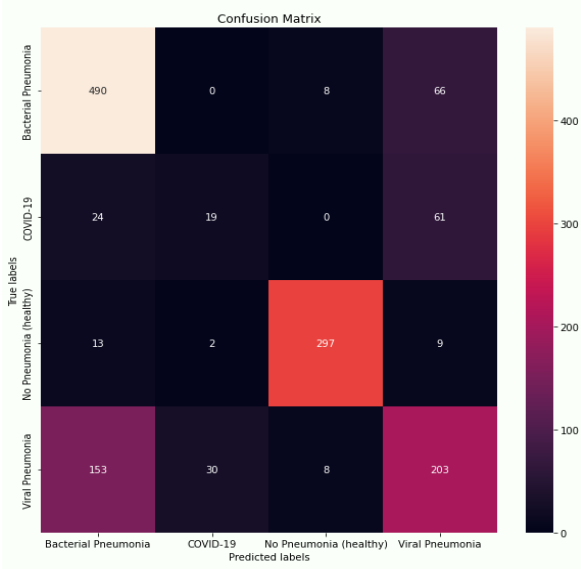

Figure 13: Validation confusion matrix for improved model

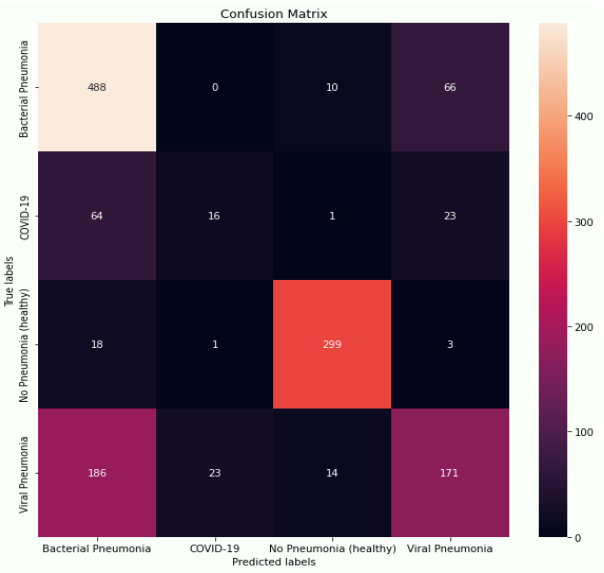

Figure 14: Test confusion matrix for improved model

5. Transfer Learning Model

In real-world problems models need to be trained from zero. Transfer learning means to use pre-trained models that were used for similar tasks by just adjusting to the specific data and task. The transfer learning model that was used is VGG16 which was introduced by (Simonyan k. et al, 2014) and it has been widely used for many image classification tasks so far with promising results. In this research we just used VGG16 to explore its capabilities, so the layers that were added to adjust the model were simple. They consist of one extra Dense layer with 128 nodes and Relu activation function, and the output dense layer with 4 nodes and the SoftMax function. The model is the largest so far in terms of parameters with 15.763.908 total parameters and the most complex, since it consists of 22 layers.

Model: "model"

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 156, 156, 3)]	0
block1_conv1 (Conv2D)	(None, 156, 156, 64)	1792
block1_conv2 (Conv2D)	(None, 156, 156, 64)	36928
block1_pool (MaxPooling2D)	(None, 78, 78, 64)	0
block2_conv1 (Conv2D)	(None, 78, 78, 128)	73856
block2_conv2 (Conv2D)	(None, 78, 78, 128)	147584
block2_pool (MaxPooling2D)	(None, 39, 39, 128)	0
block3_conv1 (Conv2D)	(None, 39, 39, 256)	295168
block3_conv2 (Conv2D)	(None, 39, 39, 256)	590080
block3_conv3 (Conv2D)	(None, 39, 39, 256)	590080
block3_pool (MaxPooling2D)	(None, 19, 19, 256)	0
block4_conv1 (Conv2D)	(None, 19, 19, 512)	1180160
block4_conv2 (Conv2D)	(None, 19, 19, 512)	2359808
block4_conv3 (Conv2D)	(None, 19, 19, 512)	2359808
block4_pool (MaxPooling2D)	(None, 9, 9, 512)	0
block5_conv1 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv2 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv3 (Conv2D)	(None, 9, 9, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 128)	1048704
dense_1 (Dense)	(None, 4)	516

Total params: 15,763,908
Trainable params: 1,049,220
Non-trainable params: 14,714,688

Figure 16: VGG16 model summary

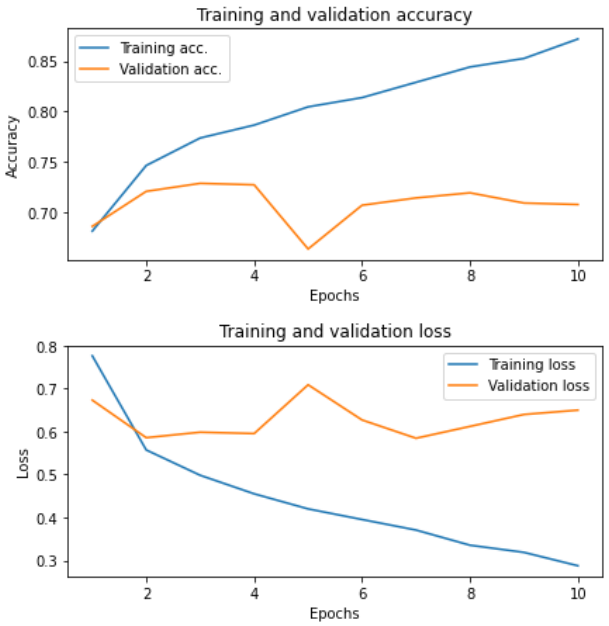

Figure 17: VGG16 train/validation accuracy & loss

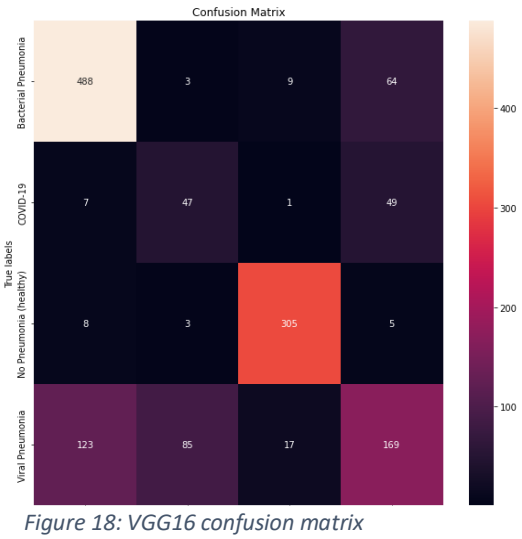

Figure 18: VGG16 confusion matrix

Even though VGG16 model have similar F1-score as the improved one, approximately 73%. It is obvious from figure 17 that the model overfits almost from the start without the use of any regularization technic or dropout layer.

Based on the confusion matrix and accompanying metrics presented in figures 18 and 19, it is evident that the class exhibiting the highest F1-score is once again no pneumonia, with a score of 93.42%. Conversely, COVID-19 was the most frequently misclassified class, with an F1-score of 33.84%. This can be attributed to the fact that COVID-19 was a minority class within the

dataset. Conversely, no pneumonia represented the majority class, resulting in higher instances and more accurate classification.

```
Class Bacterial Pneumonia:
Specificity :0.8315
Sensitivity: 0.8652
F1-score: 0.8202
Accuracy: 0.8453

Class COVID-19:
Specificity :0.9289
Sensitivity: 0.4519
F1-score: 0.3884
Accuracy: 0.893

Class No Pneumonia (healthy):
Specificity :0.9746
Sensitivity: 0.9502
F1-score: 0.9342
Accuracy: 0.9689

Class Viral Pneumonia:
Specificity :0.8807
Sensitivity: 0.4289
F1-score: 0.4963
Accuracy: 0.752
```

Figure 19: VGG16 model metrics

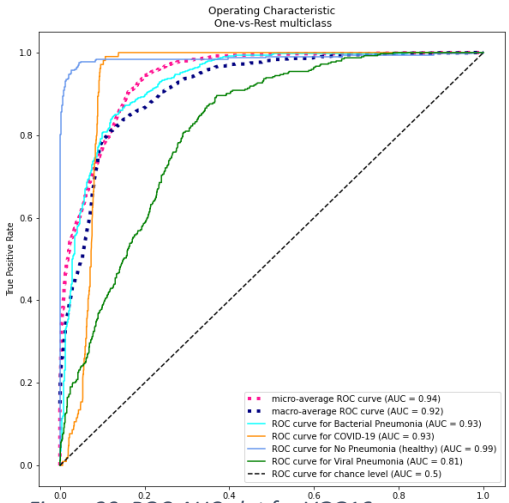

Figure 20: ROC-AUC plot for VGG16

As seen before in the other two models, similar results are shown in the ROC-AUC plot also for VGG16 model. Viral pneumonia is slightly closer to the left top corner, which can be explained from the increase in both specificity and sensitivity.

6. Discussion

In conclusion, our improved model has demonstrated a slight improvement over the baseline model in terms of average micro F1-score, by 5% and in general it performs better than the VGG16 model, because it overfits.

Additionally, our model effectively handled overfitting by implementing data augmentation and dropout layers, showing the difference that those two can achieve. However, there is still room for further improvement which could happen by increasing the model's complexity by incorporating additional dropout layers or regularization techniques in the convolutional and dense layers. Although kernel regularization was attempted in our experiment to handle overfitting, it did not outperform the effectiveness of the dropout layers.

Moreover, further improvement is necessary for VGG16 for two main reasons. Firstly, it is essential to address overfitting as it was distinct from the early stages of our experiment. Secondly, despite its complexity and prior success in similar tasks (Nyaupane, B. K., 2022), our VGG16 model did not perform as well as expected. Therefore, additional optimization is required to exploit the model's full potential.

7. References

- Reshi, A. A., Rustam, F., Mehmood, A., Alhossan, A., Alrabiah, Z., Ahmad, A., ... & Choi, G. S. (2021). An efficient CNN model for COVID-19 disease detection based on X-ray image classification. Complexity, 2021, 1-12.

- scikit-learn. (Version 1.2). Retrieved from https://scikit-learn.org/stable/index.html Retrieved 06 Mar. 23.

- Keras. (2015). Retrieved from https://keras.io/api/ Retrieved 06 Mar. 23

- Tensorflow. (Version 2.11.0). Retrieved from https://www.tensorflow.org/api_docs Retrieved 07 Mar. 23

- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Communications of the ACM 60.6 (2017): 84-90.

- Garcea, F., Serra, A., Lamberti, F., & Morra, L. (2022). Data augmentation for medical imaging: A systematic literature review. Computers in Biology and Medicine, 106391.

- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

- Nyaupane, B. K. (2022). Pneumonia of Chest X-Ray Images Detection using VGG Architectures. Journal of Lumbini Engineering College, 4(1), 43-48.