Deep Learning 880008-M-6 Assignment

Using Deep Learning to Perform Mul:-Class Classifica:on on the Covid19 Chest X-ray Dataset

Report by:

Ela Guven 2067608

March 2023

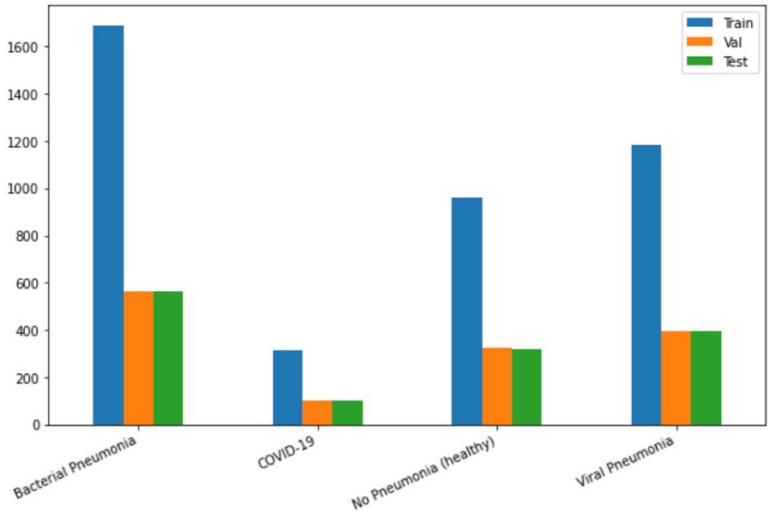
CNN Model for COVID-19 X-ray Image Classification

Convolutional Neural Network is one of the neural networks that is most studied in computer vision. With the introduction of pioneering work, AlexNet [1] CNN architecture has shown promising results in a variety of classification tasks. In addition, CNN models have generally proved their performance in a wide range of application domains. One of the domains that CNN performs significantly is medical image classification. For this assignment, a CNN algorithm was implemented for the COVID-19 X-ray dataset [2]. In this way, an automatic diagnostic system was created to detect if a person is healthy or sick by looking at chest images.

First, the required packages from the libraries are imported and data is loaded. Dataset contains 6500 chest images and 4 different categories including the COVID-19 cases. In the preprocessing step, all various images are resized by 156 x 156. In this step, the entire training set was split into training, validation, and test subsets with a proportion of 0.2 and random state of 42 by using stratified train-test splits. After this step, some visualizations were added to code to have an idea about dataset. Before the dataset was used for training model, explanatory data analysis was conducted. It was seen that the dataset was highly imbalanced regarding COVID chest images. The problem of having less COVID samples and its potential effects will be discussed in the discussion part. Since the pixel values for each image are unsigned integers, then pixel values were normalized to the floating-point numbers between 0 and 1. Then, class labels are converted to one-hot encoded vectors. In this way, prediction of images with their unique vectors will be easier.

Proportions of each 4 classes regarding training, validation and test set were plotted with their classes.

Figure 1: Class distributions over different datasets



It is seen that dataset is imbalanced and needs to be preprocessed to achieve promising results. For this moment, the model will be continued without balancing the dataset classes. In addition, there is multiclass classification problem, thus the model and activation function were designed considering this criterion.

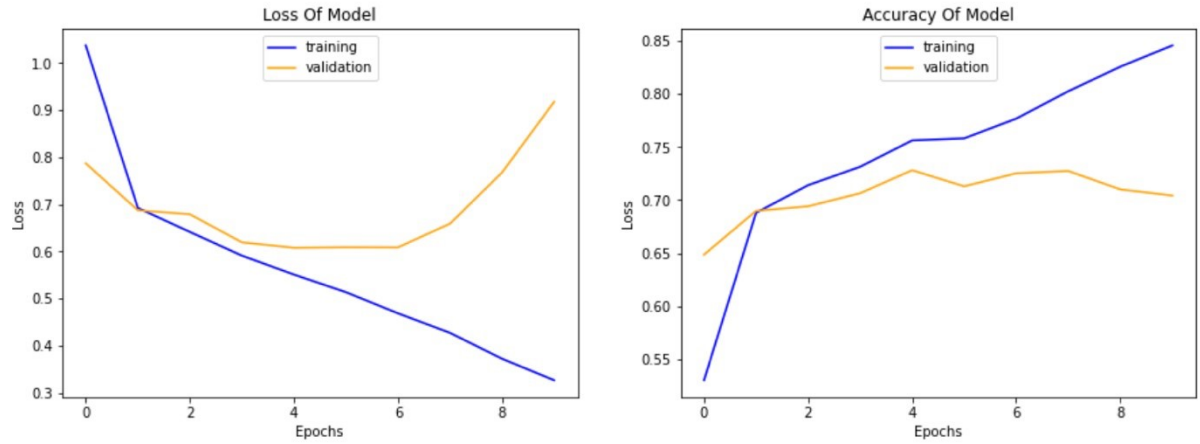
Second, the assignment-defined baseline CNN algorithm was implemented. The Keras sequential model was used to construct the network which consists of convolutional layers, max

pooling layers, dense layers, and an output layer. Convolutional layer with 64 filter with the kernel size of 3x3 with ReLu activation functions. Then, structure was followed by two times of max pooling layers of size 2x2. Features will be extracted in convolutional layers from the images and then max pooling will help to decrease the size of images while protecting the information. By flattening the layers, images are converted from two dimension to one dimension. Following this step, two dense layers of size of 32 with ReLu activation function was added. Finally, output layer size of 4 with SoftMax activation function was added to model because this is a multi-class classification problem and SoftMax is the most used activation function for these types of classification problems. This function enables to assign probabilities for images in the classes. As it is stated in the description, Adam as optimizer and the accuracy as metric were utilized for this model. Cross entropy was selected for loss function during the optimization. The baseline model was trained with 10 epochs and batch size as 32.

Table 1: Baseline CNN model for image classification

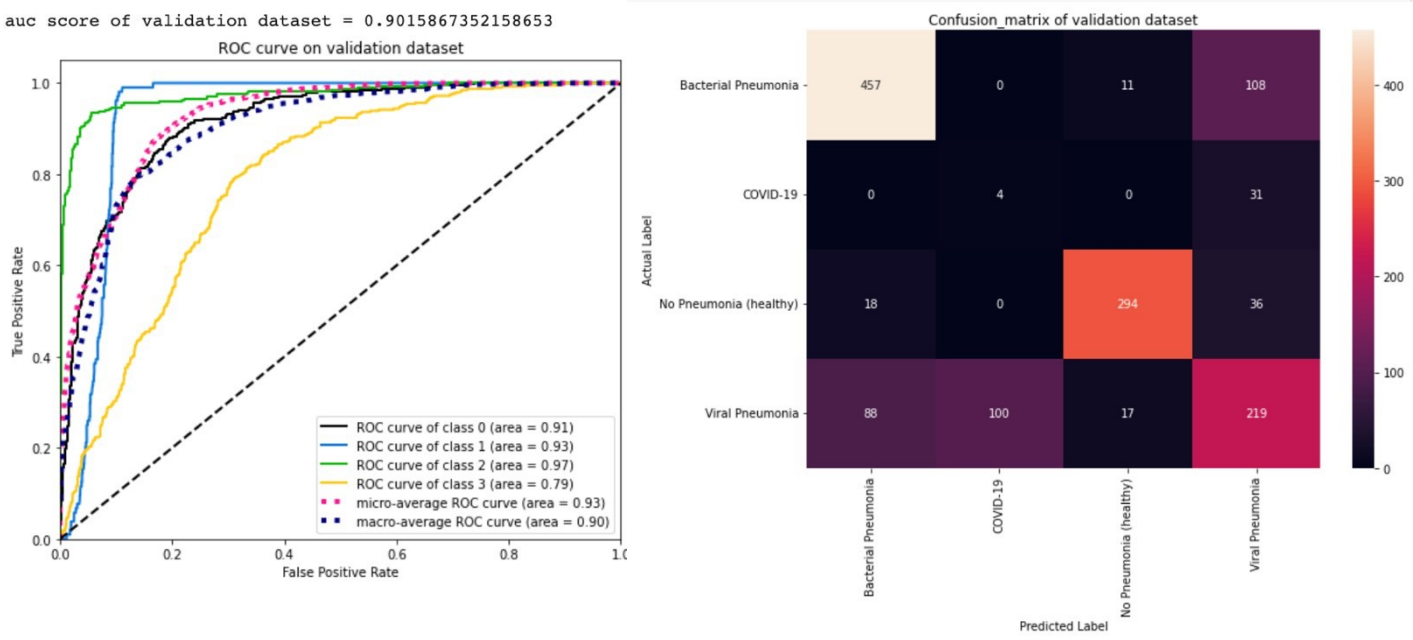
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 156, 156, 64)      1792

 conv2d_1 (Conv2D)           (None, 156, 156, 32)      18464

 max_pooling2d (MaxPooling2D  (None, 78, 78, 32)       0
 )

 conv2d_2 (Conv2D)           (None, 78, 78, 64)        18496

 conv2d_3 (Conv2D)           (None, 78, 78, 32)        18464

 max_pooling2d_1 (MaxPooling  (None, 39, 39, 32)       0
 2D)

 flatten (Flatten)           (None, 48672)             0

 dense (Dense)               (None, 32)                1557536

 dense_1 (Dense)             (None, 32)                1056

 dense_2 (Dense)             (None, 4)                 132

=================================================================
Total params: 1,615,940
Trainable params: 1,615,940
Non-trainable params: 0
_________________________________________________________________
```

Figure 2: Accuracy and loss plots



These two plots show that training accuracy is going to be higher than validation accuracy. This means that the performance of the model on the training dataset has better performance than validation. When the loss of model is analyzed, loss in the training is getting less over time but the loss on the validation is not. The problem of overfitting can be prevented by utilizing early stopping or other hyperparameter tunings in the improved model.

Figure 3: ROC curve and confusion matrix of validation datasets



a. ROC Curve b. Confusion Matrix

ROC curves show the model performance by examining the true positive rates and false positive rates. If the curves are closer to the top left corner, then the model giving better performance for that class. In the confusion matrix, model didn't fully classify the classes well because there are many wrong classifications for bacterial pneumonia. Also, model wasn't successful to detect the covid samples. Since there wasn't enough Covid class in the training process, model didn't learn much.

Table 2: Performance metrics of validation dataset

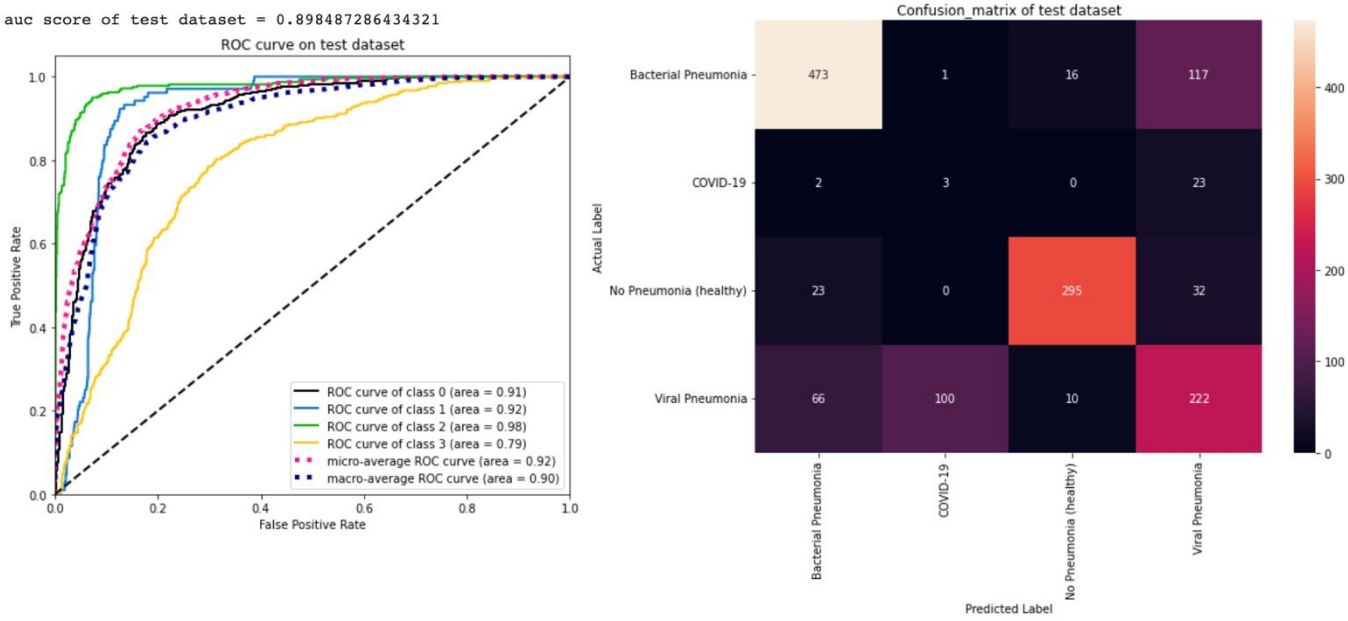
Classification_report of validation dataset

	precision	recall	f1-score	support
0	0.79	0.81	0.80	563
1	0.11	0.04	0.06	104
2	0.84	0.91	0.88	322
3	0.52	0.56	0.54	394
accuracy			0.70	1383
macro avg	0.57	0.58	0.57	1383
weighted avg	0.68	0.70	0.69	1383

Specificity Val set

Bacterial Pneumonia: specificity = 0.8686
COVID-19: specificity = 0.9258
No Pneumonia (healthy): specificity = 0.9729
Viral Pneumonia: specificity = 0.8175

Figure 4: ROC curve and confusion matrix of test dataset



a. ROC Curve

b. Confusion Matrix

Table 3: Performance metrics of test set

Classification_report of test dataset

	precision	recall	f1-score	support
0	0.78	0.84	0.81	564
1	0.11	0.03	0.05	104
2	0.84	0.92	0.88	321
3	0.56	0.56	0.56	394
accuracy			0.72	1383
macro avg	0.57	0.59	0.57	1383
weighted avg	0.68	0.72	0.70	1383

Specificity Test set

Bacterial Pneumonia: specificity = 0.8827
COVID-19: specificity = 0.9255
No Pneumonia (healthy): specificity = 0.9748
Viral Pneumonia: specificity = 0.8254

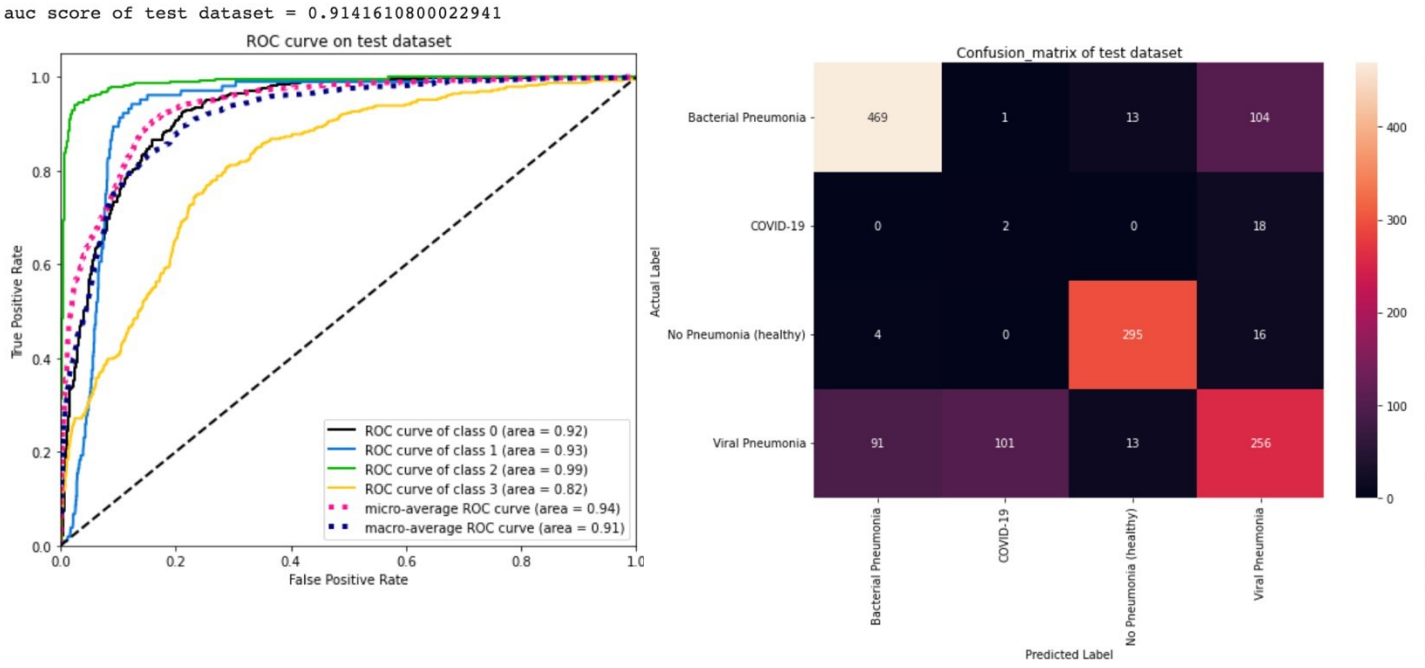
An improved model was designed considering the baseline model and the aim was to increase the performance of model on the test set. In the hyperparameter tuning, the most effective parameters were selected to create an improved CNN model. The problem of overfitting from the baseline model were tried to reduce in this model. For this purpose, regularization, drop out were added to model. Regularization is one of the most effective techniques to prevent overfitting and decrease the complexity of the model [3]. L2 regularization was selected because this enables to reach out important features from the images. Dropout was also used with 30% dropout rate to remove randomly a different subset of neurons on the layer during the training process. In this way, smaller network was created with each epoch on the training set. Therefore, more robustness to image variations was obtained. Another way to reduce the network size was to reduce the number of neurons in the dense layer from 32 to 16 after flattening layers. In the improved model, the number of epochs as 20 and batch size as 32 were used. Furthermore, learning rate was decreased by 0.2 factor in case of no improvement on the validation loss.

Data augmentation techniques are effective to increase the performance of the model [4]. It helps to generate more data from existing dataset. In this way, model will learn more features. In data augmentation part, flipping, rotating, and cropping were tried in the preprocessing step. Even though dataset was extended with these transformed images from existing images, model accuracy of CNN model wasn't improved much. Therefore, group members didn't want to include this part on the code. Implementing the data augmentation techniques extensively would attain to better accuracy and performance but limited size of dataset and imbalanced class distribution have still affected the model results.

With the improved model, overfitting problem has been resolved in small amount. Unfortunately, interclass confusion remains the same. Classifier hasn't still recognized the Covid classes for most of the time. Also, some deteriorations have been observed such as

more wrong classification in the bacterial class. Therefore, hyperparameter tunning is not enough to improve the network itself.

Figure 5: ROC curve and confusion matrix of test dataset with improved model



a. ROC Curve

b. Confusion Matrix

Table 4: Performance metrics of test dataset with improved model

Classification_report of test dataset

	precision	recall	f1-score	support
0	0.80	0.83	0.81	564
1	0.10	0.02	0.03	104
2	0.94	0.92	0.93	321
3	0.56	0.65	0.60	394
accuracy			0.74	1383
macro avg	0.60	0.60	0.59	1383
weighted avg	0.71	0.74	0.72	1383

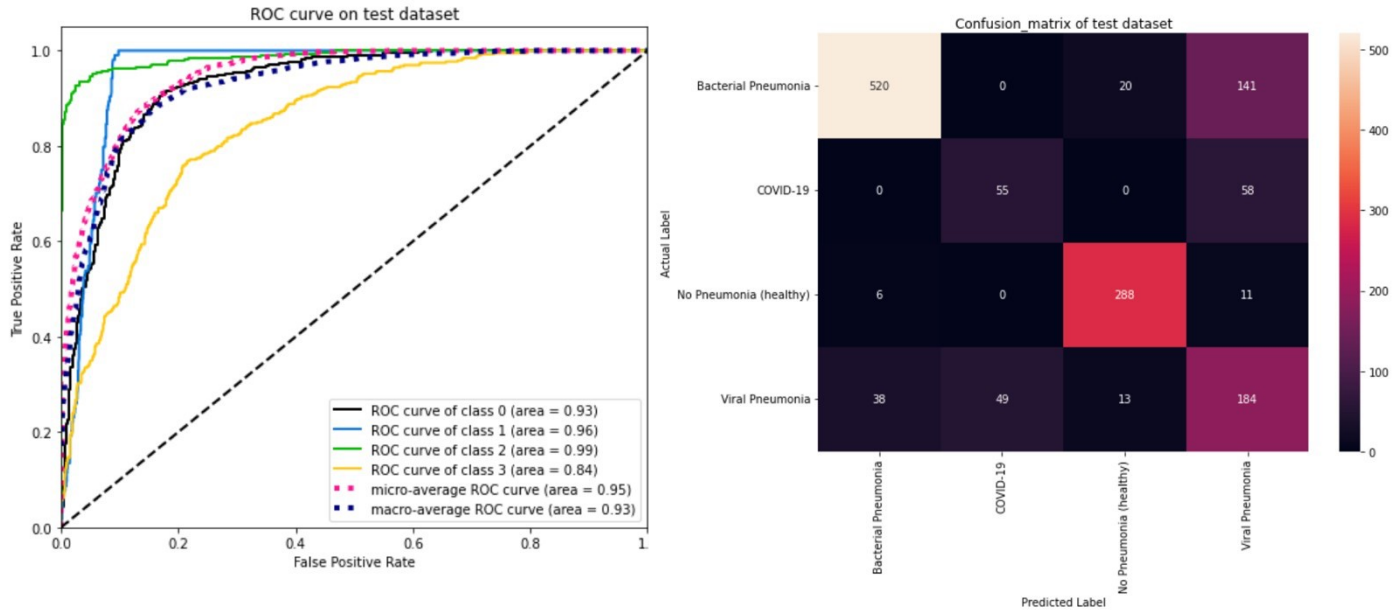
Specificity Val set

Bacterial Pneumonia: specificity = 0.8807
COVID-19: specificity = 0.9252
No Pneumonia (healthy): specificity = 0.9757
Viral Pneumonia: specificity = 0.8503

As last step, the model was trained using transfer learning and pattern from VGG16 model was used to detect the classes. Compared to baseline and improved model, this model has achieved best performance. This pretrained model enables to learn more from the images and increase performance of the model. Using this model improved to detect some classes such as bacterial one but rest of the classes wasn't recognized by the model. Potential reason for this result is highly imbalanced data. To overcome this issue, dataset needed to be preprocessed with different balancing technique.

Figure 6: ROC curve and confusion matrix of test set with transfer learning model

auc score of test dataset = 0.9282131603438034



a. ROC Curve

b. Confusion Matrix

Table 5: Performance metrics of test set with transfer learning model

Classification_report of test dataset

	precision	recall	f1-score	support
0	0.76	0.92	0.84	564
1	0.49	0.53	0.51	104
2	0.94	0.90	0.92	321
3	0.65	0.47	0.54	394
accuracy			0.76	1383
macro avg	0.71	0.70	0.70	1383
weighted avg	0.75	0.76	0.75	1383

Specificity Val set

Bacterial Pneumonia: specificity = 0.9373
COVID-19: specificity = 0.9614
No Pneumonia (healthy): specificity = 0.9694
Viral Pneumonia: specificity = 0.8089

References

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

2. COVID-19 X-ray dataset: https://darwin.v7labs.com/v7-labs/covid-19-chest-x-ray-dataset?sort=priority% 3Adesc.

3. Park S. and Kwak N. “Analysis on the Dropout Effect in Convolutional Neural Networks”

4. C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” Journal of Big Data, vol. 6, p. 60, 2019.

5. D. Ho, E. Liang, and R. Liaw, “1000x Faster Data Augmen- tation, Berkeley Artificial Intelligence Research,” Berkeley, CA, USA, 2019, https://bair.berkeley.edu/blog/2019/06/07/ data_aug/.

6. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. “Dropout: A

Simple Way to Prevent Neural Networks from Overfitting.” Journal of Machine Learning Research 15 (2014) 1929-1958