Grouped CNN Performance

In this model, the input is split along the channel axis, which results in 3 groups. Each group is convolved separately and at last concatenated as the output. As the settings required that the channels and filters to be divisible by the number of groups, the first layer uses 36 filters whereas second layer uses 66 filters. Thus, each group will be convolved with 13 and 22 filters respectively. Due to the difference in number of filters, the number of trainable parameters has also increased to 628,485 parameters, which is by far the highest in all three CNN models. Hence, the required training time was longer.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resizing (Resizing)         (32, 224, 224, 3)         0

 conv2d (Conv2D)             (32, 224, 224, 36)        360

 max_pooling2d (MaxPooling2D  (32, 112, 112, 36)       0
 )

 activation (Activation)     (32, 112, 112, 36)        0

 conv2d_1 (Conv2D)           (32, 112, 112, 66)        7194

 max_pooling2d_1 (MaxPooling  (32, 56, 56, 66)         0
 2D)

 activation_1 (Activation)   (32, 56, 56, 66)          0

 flatten (Flatten)           (32, 206976)              0

 dense (Dense)               (32, 3)                   620931

=================================================================
Total params: 628,485
Trainable params: 628,485
Non-trainable params: 0
```

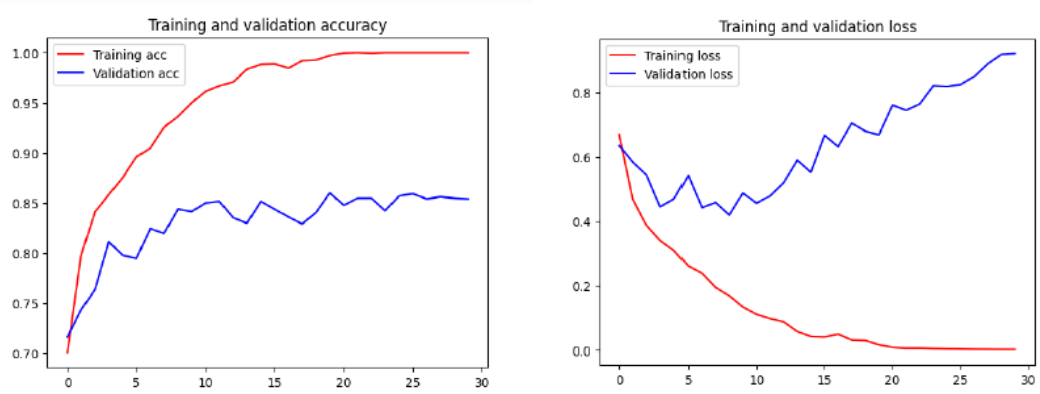


Figure 25 Accuracy and Loss of training and validation set for Grouped CNN base model.

```
Epoch 30/30
238/238 [==============================] - 421s 2s/step - loss: 7.1807e-04 - accuracy: 1.0000 - auc: 1.0000 - val_loss: 0.9226 - val_accuracy: 0.8538
```

overfitting is significant. The accuracy of validation stays stagnant from epoch 8 to 30 despite small fluctuations in between. In contrast, validation decreases from epoch 0 to 3, fluctuates between epoch

5 to 10 and increases gradually to 0.9226 at epoch=30. This model does not generalize well on unseen data even though training accuracy and loss showed good performance.

Regularization

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resizing_1 (Resizing)       (None, 90, 90, 3)         0

 conv2d_2 (Conv2D)           (None, 90, 90, 36)        360

 max_pooling2d_2 (MaxPoolin  (None, 45, 45, 36)        0
 g2D)

 activation_2 (Activation)   (None, 45, 45, 36)        0

 dropout_2 (Dropout)         (None, 45, 45, 36)        0

 batch_normalization_2 (Bat  (None, 45, 45, 36)        144
 chNormalization)

 conv2d_3 (Conv2D)           (None, 45, 45, 66)        7194

 max_pooling2d_3 (MaxPoolin  (None, 22, 22, 66)        0
 g2D)

 activation_3 (Activation)   (None, 22, 22, 66)        0

 dropout_3 (Dropout)         (None, 22, 22, 66)        0

 batch_normalization_3 (Bat  (None, 22, 22, 66)        264
 chNormalization)

 flatten_1 (Flatten)         (None, 31944)             0

 dense_1 (Dense)             (None, 3)                 95835

=================================================================
Total params: 103797 (405.46 KB)
Trainable params: 103593 (404.66 KB)
Non-trainable params: 204 (816.00 Byte)
```

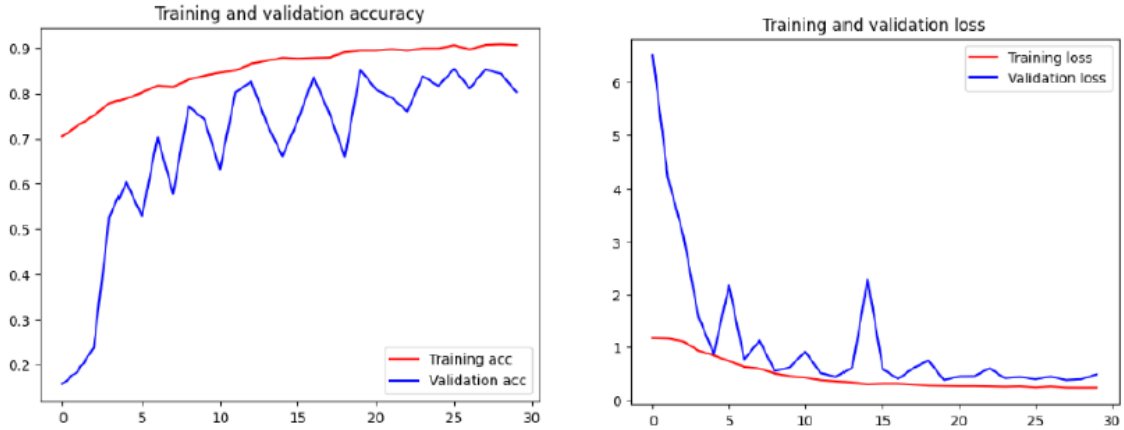


Figure 26 Accuracy and Loss for Grouped CNN Regularized Model

It is observed that overfitting is resolved after regularization. Both training and validation accuracy as well as loss moves towards the same direction. Throughout the 30 epochs, the training accuracy and loss improve gradually. On the other hand, there is a sharp increase in validation accuracy between epoch 0 to 3. It is observed that the validation loss experienced a steep fall at epoch 1, in which the loss in validation set approaches 0.4886 at the end of training. At epoch=30, the training accuracy was 90.70% with 0.2317 loss, whereas for validation set, the accuracy was 80.33% with 0.4886 loss.

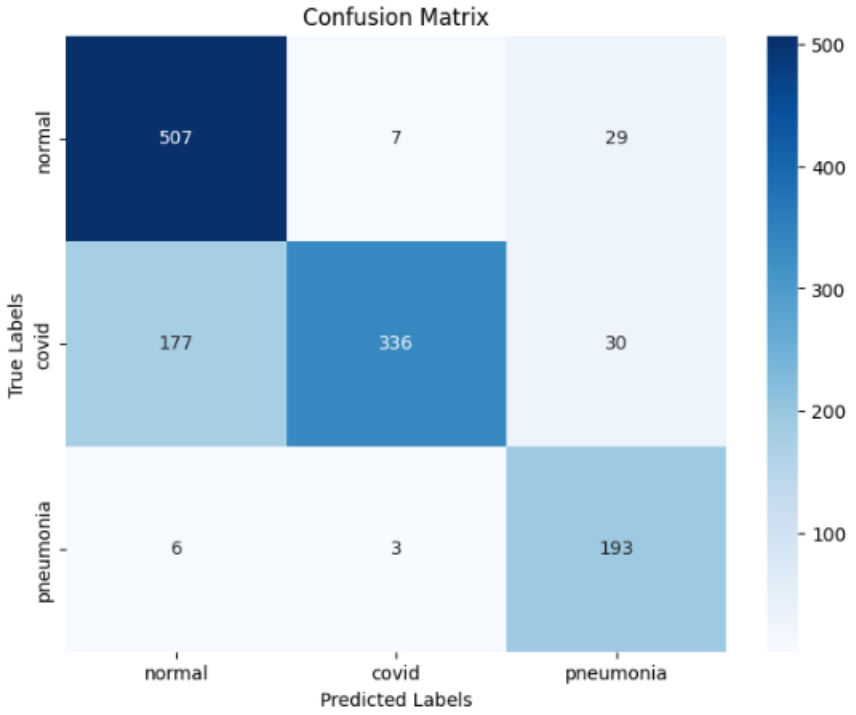
Test Set

```
:  1 #evaluate on test set
   2 test_loss, test_accuracy, test_auc = gcnn_reg.evaluate(test_generator)

41/41 [==============================] - 5s 109ms/step - loss: 0.4941 - accuracy: 0.8043 - auc: 0.9361
```

The model was further evaluated on test set with 1288 images, the model achieved 80.43% accuracy with 0.4941 loss and 0.9361 AUC.

```
:
1  import matplotlib.pyplot as plt
2  import seaborn as sns
3
4  # Visualize the confusion matrix
5  plt.figure(figsize=(8, 6))
6  sns.heatmap(confusion_gcnn, annot=True, fmt='d', cmap='Blues',xticklabels=classLst, yticklabels=classLst)
7  plt.xlabel('Predicted Labels')
8  plt.ylabel('True Labels')
9  plt.title('Confusion Matrix')
10 plt.show()
```



Based on the confusion matrix, the model correctly classifies 507 normal, 336 Covid-19 and 193 pneumonia CXR. On the contrary, 36 normal, 207 covid and 9 pneumonia CXR were misclassified. Similarly, it is observed that the model struggles to differentiate between normal and Covid-19 CXR.

	precision	recall	f1-score	support
normal	0.73	0.93	0.82	543
covid	0.97	0.62	0.76	543
pneumonia	0.77	0.96	0.85	202
accuracy			0.80	1288
macro avg	0.82	0.84	0.81	1288
weighted avg	0.84	0.80	0.80	1288

The macro average precision, recall and f1-score are 0.82, 0.84 and 0.81 respectively. Among all classes, the model shows the highest precision in covid class while highest recall in pneumonia class. In other words, when the model classifies the CXR as normal, it is correct 97% of the time. Besides that, the model correctly identifies 96% of all pneumonia CXR. The macro average f1-score which consider all classes as equally important is close to 1.