CS 577: Deep Learning

Final Project Report

Prof. Gady Agam

Image Classification to predict pneumonia on chest X-Ray images using CNN

Submitted by: **Parth Gupta (A20449774)**

Shirish Vogga (A20456808)

Abstract / Introduction:

Pneumonia is a disease which occurs in the lungs and through early diagnosis is effectively treatable. Usually, it is diagnosed through chest X-Ray images by radiologists. The diagnosis of this disease may vary and is subjective. Since, there exists no fundamental detection mechanism, conclusion based on reading the X-rays can be True positive and false positive. Thus, a computer-aided technology is required to make the detection full-proof. This is where, our study comes in. We implement 3 models to evaluate pneumonia cases. We trained a custom CNN neural network, implement transfer learning using VGG16 and Xception networks, to draw conclusions and predict which model gives us the best results. We achieve, an accuracy of 92% on VGG16 model and 90% accuracy on the Custom model, and even though the accuracy is higher on Custom network, we analyse that the model better predicts using VGG16.

Recent studies have shown that using CNN has garnered accurate and successful results. Convolutional Neural networks have the edge over standard deep networks as they have the ability to extract significantly larger features from the complete image rather than handcrafted features. CNN's have better prediction results on medical classification such as: Breast cancer detection, Brain Tumour detection, etc. To fully optimize our models and avoid overfitting / vanishing gradient problems, we have performed regularization techniques, hyper tuned the hyper parameters, performed data augmentation methods.

Problem statement / Question:

A. Data

The dataset consists of 5,862 images provided from Kaggle. The images are chest X-rays containing 2 classes: Normal and Pneumonia images.

The data provided to us was an imbalanced dataset with 4000 images in train set, 1216 images in the validation set and 624 images in the test set. These images were not split evenly such as, there were 1000 images in the train set for category normal cases and 3000 images in the train set for category pneumonia cases.

We then created a balanced dataset with 2000 images in the train set and 1000 images for each category: normal and pneumonia. For validation set, we had 341 images and test set contains 234 normal images and 390 pneumonia images.

These images are in the resolution range from: 712x439 to 2338x2025. In our model, we represent 0 for normal cases and 1 for pneumonia cases.

For Imbalanced Set

	Train	Validation	Test
Normal	1000	341	234
Pneumonia	3000	875	390
Total	4000	1216	624

For Balanced Set

	Train	Validation	Test
Normal	1000	341	234
Pneumonia	1000	341	390
Total	2000	682	624

B. Statement

The final objective of the project is to predict whether the given image is a pneumonia case or not.

Balanced Set:

The final evaluation metric we used is accuracy, precision, recall and f1 score. We have trained our own custom CNN model and evaluated test accuracy of 90%.

Next, we train on VGG16 model to obtain an accuracy of 92%, precision value of 92%, recall value of 94%, and f1-score of 94%.

Lastly, we evaluate for the Xception model to record an accuracy of 65%, a precision value of 81%, recall score of 8% and f1-score of 14%.

Here, we show the results:

Balanced Dataset			
	Precision (%)	Recall (%)	F1-score (%)
	Custom		
Normal Class	92	79	85
Pneumonia Class	89	96	92
Accuracy (%)	90		
	VGG16		
Normal Class	91	86	89
Pneumonia Class	92	95	94
Accuracy (%)	92		
	Xception		
Normal Class	39	97	55
Pneumonia Class	81	8	14
Accuracy (%)	41		

Imbalanced Dataset			
	Precision (%)	Recall (%)	F1-score (%)
	Custom		
Normal Class	92	79	85
Pneumonia Class	89	96	92
Accuracy (%)	90		
	VGG16		
Normal Class	76	97	85
Pneumonia Class	98	82	89
Accuracy (%)	87		
	Xception		
Normal Class	81	66	73
Pneumonia Class	82	91	86
Accuracy (%)	82		

Here, we use Transfer Learning for best prediction of our model and perform a comparison between different transfer learning models and a custom CNN model. Transfer learning is a concept that allows pre-trained networks to be utilized with a custom dense model. The weights of the CNN layers are stored on the pre-trained model that has been trained on a very large dataset of more than 100,000 images on ImageNet. These CNN layers are then frozen to avoid re-learning of weights and some dense layers are added in the model. This technique allows a better result while training the model. Thus, transfer learning was used by us.

Problem Approach / Methods of solving:

1. Data Analysis:
    - We first analyse the data, predictor variable and label. We conclude, we have 2 labels representing normal and pneumonia cases.
    - The dataset downloaded is imbalanced and each category has different number of images in each set (train, validation and test). The data is divided into train and test set in approximately 80:20 split.
    - We represent 0 with normal cases and 1 with pneumonia cases.
2. Data Generator:
    - To implement and fit the model, we first create sub directories for our data generator. This function is required to fit the model and thus perform prediction. We create train, validation and test directories, each with 2 sub directories of normal and pneumonia images. Since, the images are not included in the keras package, we downloaded this dataset externally (Kaggle) and to predict the model, train, validation and test data generators are required.
3. Normalization of Pixels:
    - Normalization is a standard practice in Deep Learning to avoid overfitting and perform convergence faster with better accuracy. We normalized the entire train, validation and test sets with a base of 255 which is the maximum value that a pixel can have.
4. Model creation:
    - We first build a fully custom CNN neural network. We use 5 convolutional layers and 2 dense layers. We flatten the layers after the last convolutional layer and before the first

dense layer. We experimented with the number of dense layers and the number of neurons in each layer and found that 2 dense layers gave us an optimum result and any further addition of layers did not exhibit improvement. For the output layer, the activation function used is 'Sigmoid', this is because the classification is a binary classification problem with only 2 labels. In between the convolutional layers, we added maxpool layers to decrease the dimensions for faster convergence. The final metric for performance is 'Accuracy' and the loss function chosen is 'Binary cross entropy loss' which is the best loss function for binary classification tasks. The optimizer we chose is 'Adam' as it includes the properties of both, 'Relu' and 'RMSprop', along with a learning rate of 1e-4 which were selected based on hyper tuning.

- Next, we used transfer learning to build a model using VGG16. The input images we use is of dimension 244x244x3 and batch size of 20. We freeze all the convolutional layers in the VGG16 model and get an accuracy of 92.5%. Similar to the custom model, we use 2 dense layers and performed regularization as necessary.
- We then train a model with the last 5 layers unfrozen and the rest frozen. We got an accuracy of 92% with a precision value of 92 and a f1 value of 94.
- Finally, we create a model using Xception transfer learning. The input image size is 299x299x3 instead of 244x244x3 of the VGG16, this is because the pre trained images had a size of 244x244 in the case of VGG16 while for Xception the pretrained images had an input size of 299x299. We get a precision value of 81% for true cases of pneumonia. Similar to the Custom model, we hyper tuned, performed data augmentation and regularization such as number of dense layers, learning rate, batch normalization and more.

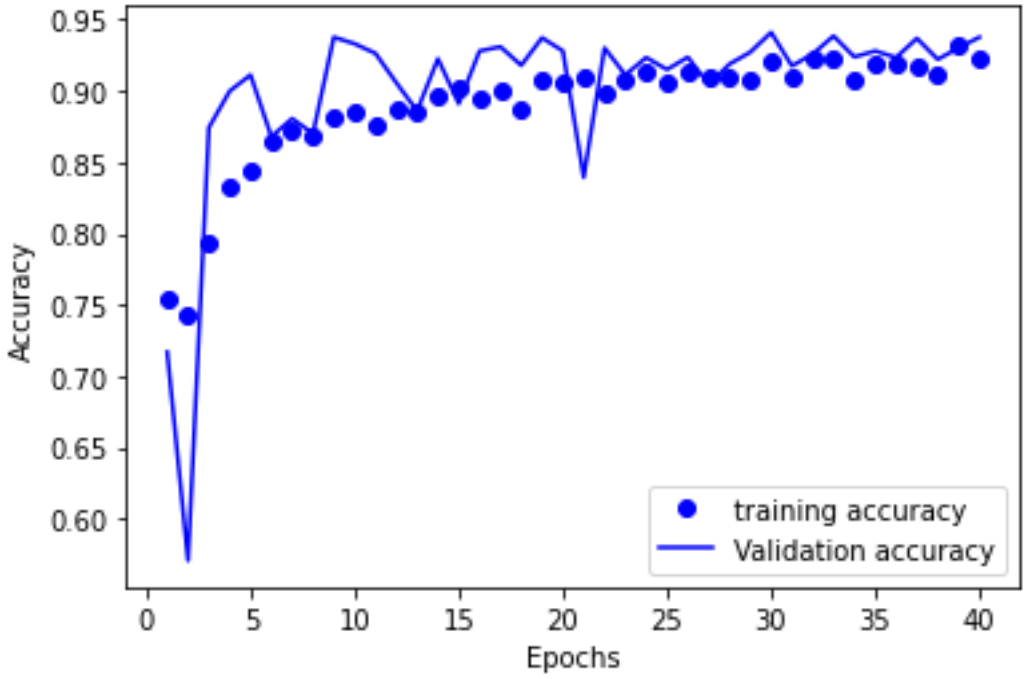
5. Training and testing the model:
    - We train the custom, VGG16 and the Xception model for 40 epochs with a batch size of 20.
6. Model save:
    - We have saved each model to save the weights for future use.

Results:

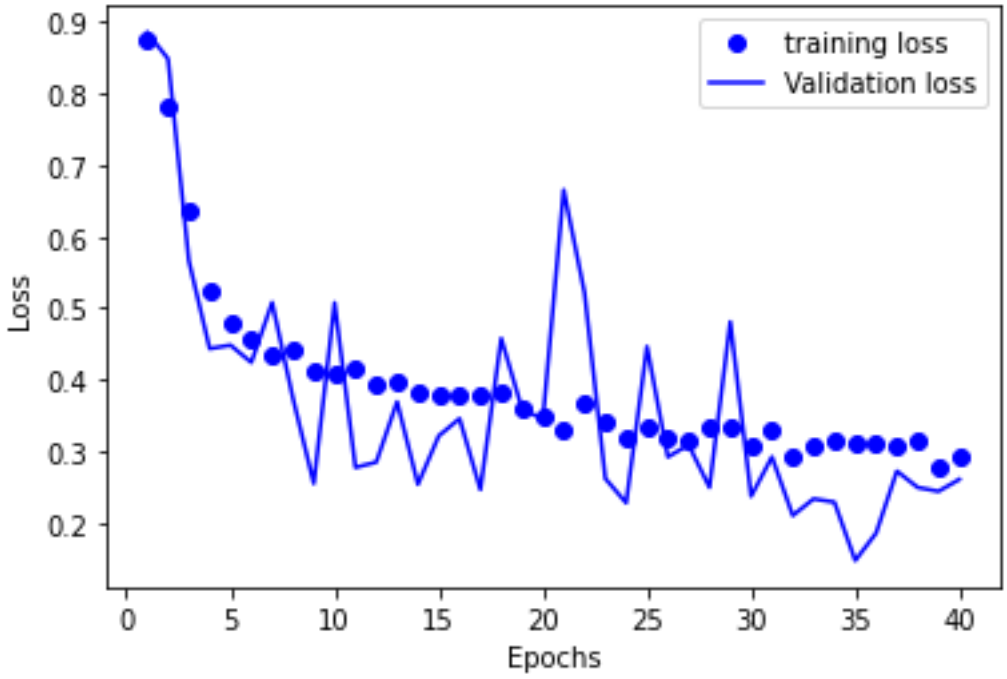
The following are the plots for accuracy and loss for each of the models trained on both the balanced and the imbalanced sets.

Imbalanced Dataset:

Custom CNN model plot for Accuracy vs Epochs



Custom CNN model for Loss vs Epochs



Here we plot the Summary for Custom CNN model:

```
26/26 [==============================] - 6s 229ms/step
[[186  48]
 [ 17 373]]
Report :
              precision    recall  f1-score   support

       False       0.92      0.79      0.85       234
        True       0.89      0.96      0.92       390

    accuracy                           0.90       624
   macro avg       0.90      0.88      0.89       624
weighted avg       0.90      0.90      0.89       624
```

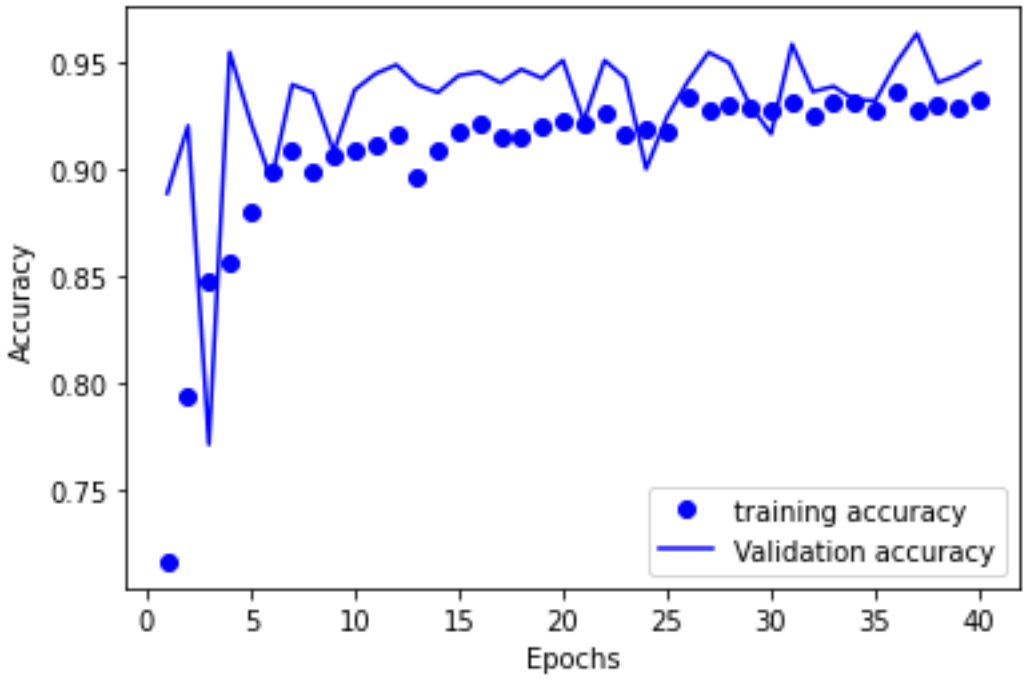
As can be seen, for the first plot Accuracy vs Epochs, our model consistently trains well with increasing epochs and the validation accuracy increases with the training accuracy. Thus, there is no overfitting in our model and our model predicts well.

Also, in the second plot, Loss vs Epochs, the validation loss is consistently decreasing with the training loss which complements the accuracy vs epochs plot and thus, a conclusion can be drawn with no overfitting and our model predicting well. A final test accuracy of 90% is achieved.

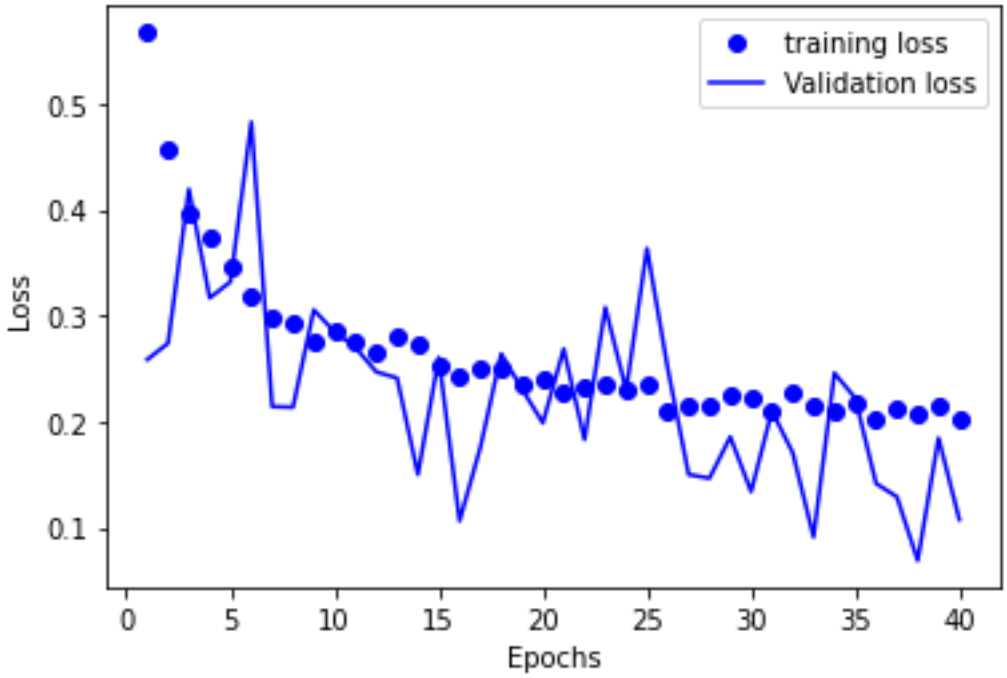
In the third graph, we observe a good precision value of 89% and a recall value of 96% with 92% f1-score. Overall, we can conclude that our model behaves as expected with a good performance.

For VGG16 Frozen model, we observe:

Accuracy vs Epochs



For Loss vs Epochs



Summary plot for Frozen VGG16 model

```
26/26 [==============================] - 6s 244ms/step
[[227   7]
 [ 72 318]]
Report :
              precision    recall  f1-score   support

       False       0.76      0.97      0.85       234
        True       0.98      0.82      0.89       390

    accuracy                           0.87       624
   macro avg       0.87      0.89      0.87       624
weighted avg       0.90      0.87      0.88       624
```

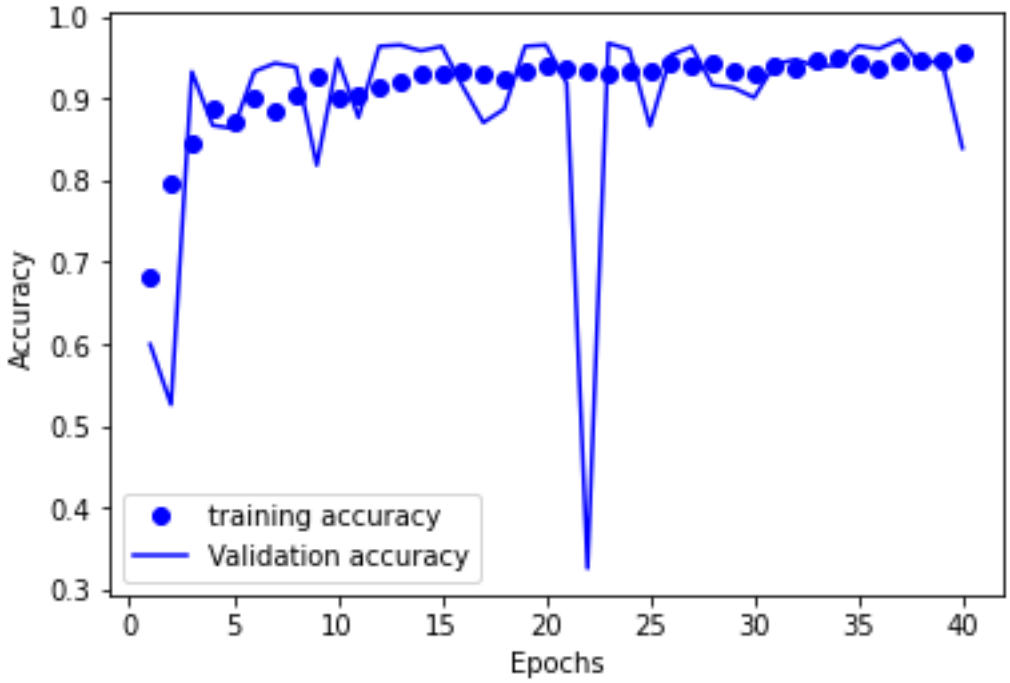
In the first plot: Accuracy vs Epochs, we conclude a validation accuracy of 95% which consistently increases with the training accuracy. There is a sudden decline in the validation accuracy within the first 5 epochs, this maybe due to randomness in the data which is expected.

Next, the Loss vs epochs plot consistently decreases with the training loss without much fluctuations. This further proves that our model is learning well and converging as expected.

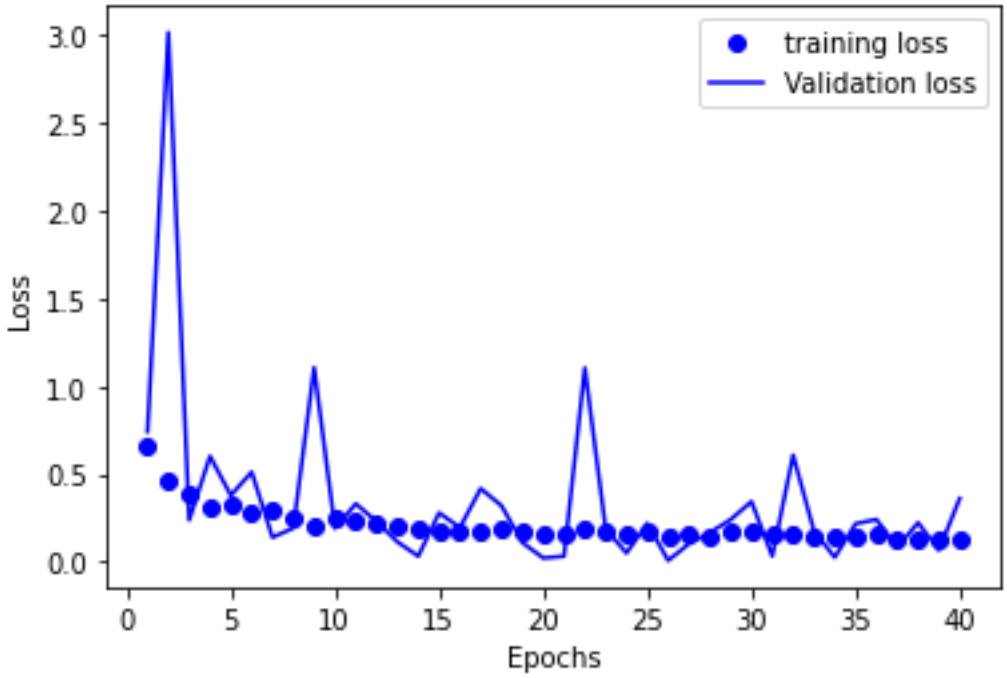
In the plot summary, we can see a final test accuracy of 87% with a true precision score of 98% and a f1-score of 89%. We conclude that this model is thus performing very well and gives us much better results than the custom CNN model, we had trained previously.

For VGG16 Unfrozen model:

Accuracy vs Epochs



For Loss vs Epochs



Here we can conclude that the model predicts well and as the training accuracy increases, the validation accuracy also increases consistently with a sudden decline at about epoch 22, this maybe due to the fact that the model predicted a wrong example and since then continued to predict accurately. As the last epoch approaches the validation accuracy decreases, this maybe because we are overfitting the model after 40 epochs.

The loss graphs also show a consistent decrease with increasing epochs and hence our model is training well.

We conclude the validation accuracy of 85%.

Xception Model:

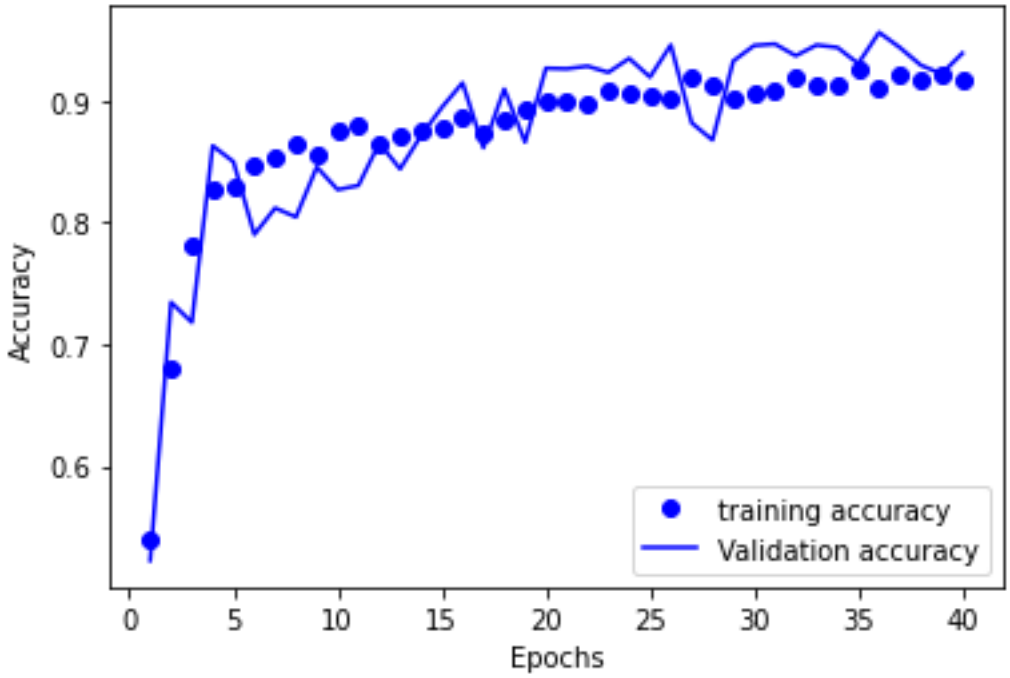
```
26/26 [==============================] - 8s 312ms/step
[[155  79]
 [ 36 354]]
Report :
              precision    recall  f1-score   support

       False       0.81      0.66      0.73       234
        True       0.82      0.91      0.86       390

    accuracy                           0.82       624
   macro avg       0.81      0.79      0.79       624
weighted avg       0.82      0.82      0.81       624
```

For the Xception model, we get a final accuracy of 82% with a precision value of 82%, recall value of 91% and f1-score 86%. While the precision value for the VGG model is 98%.

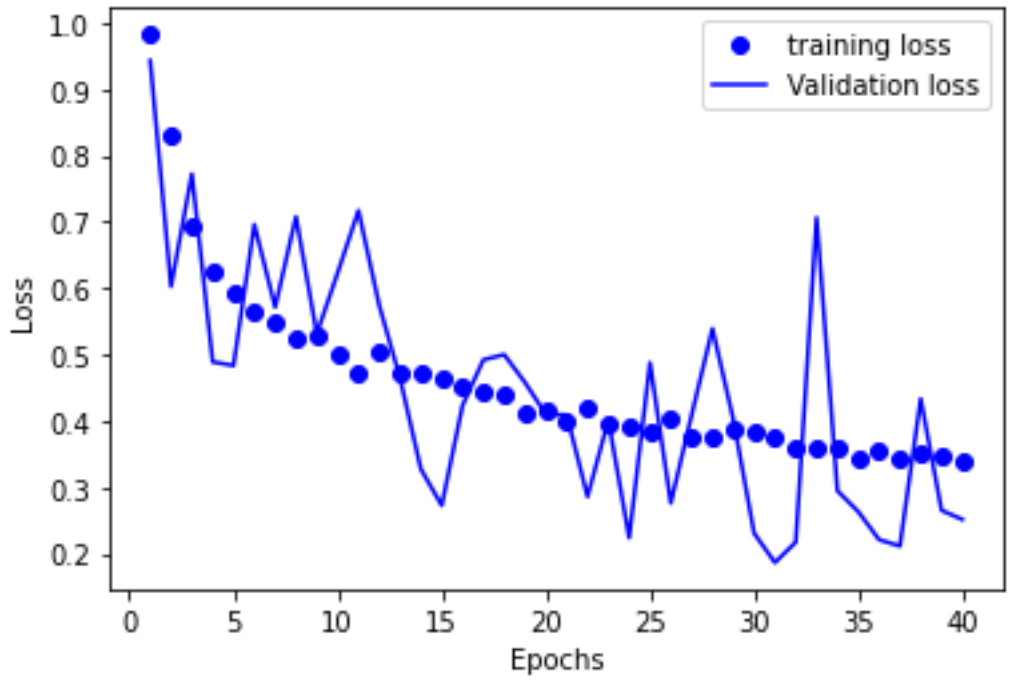
Therefore, we conclude that the VGG16 mode outperforms on our dataset and thus a 98% precision value can be concluded.

For Balanced Set:

Accuracy vs Epochs for Custom CNN network



Loss vs Epochs plot for Custom CNN network



Summary for VGG16 Unfrozen layers

```
26/26 [==============================] - 6s 236ms/step
[[202  32]
 [ 19 371]]
Report :
              precision    recall  f1-score   support

       False       0.91      0.86      0.89       234
        True       0.92      0.95      0.94       390

    accuracy                           0.92       624
   macro avg       0.92      0.91      0.91       624
weighted avg       0.92      0.92      0.92       624
```

Summary for Xception model:

```
0.4118589758872986 0.776775062084198
26/26 [==============================] - 8s 313ms/step
[[227   7]
 [360  30]]
Report :
              precision    recall  f1-score   support

       False       0.39      0.97      0.55       234
        True       0.81      0.08      0.14       390

    accuracy                           0.41       624
   macro avg       0.60      0.52      0.35       624
weighted avg       0.65      0.41      0.30       624
```

In the first plot for Accuracy vs Epochs, we see a gradual increase in validation accuracy with respect to the training accuracy as the epochs increase. This means that our custom model is training well without overfitting and hence gives us a good prediction.

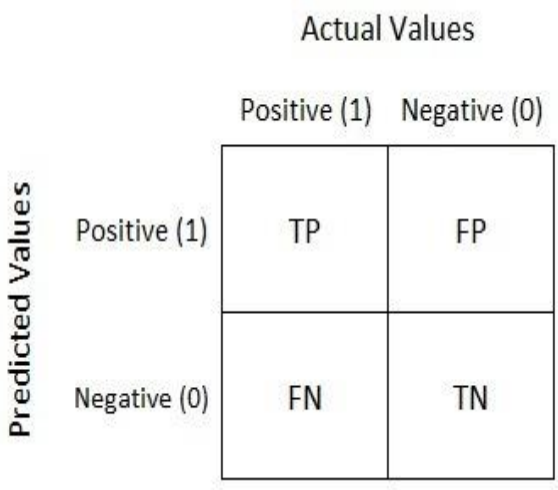
The loss plot also proves that our model is fitting well as there is a gradual decrease in validation loss with the training loss. We conclude a final accuracy of 90%.

In the last two plots, we conclude the summary for the **VGG16 Unfrozen** model to draw a comparison with the custom model and the Xception model.

We can conclude that the VGG16 model gives us a much better precision value of 92% and a much better f1-score of 94% when compared with the Xception mode.

Confusion Matrix:

Here we can observe the true positive and true negative values. The yellow marked arrow signifies the confusion matrix for each of the model. Adding the diagonal elements gives us the true values of that class. In this case, we got 186+373 true value for the pneumonia cases and the 48+17 values gives us the normal cases.




```
26/26 [==============================] - 6s 229ms/step
[[186  48]
 [ 17 373]]
Report :
              precision    recall  f1-score   support

       False       0.92      0.79      0.85       234
        True       0.89      0.96      0.92       390

    accuracy                           0.90       624
   macro avg       0.90      0.88      0.89       624
weighted avg       0.90      0.90      0.89       624
```

Conclusion:

We compared 2 datasets, Balanced and Imbalanced and build custom CNN models on both. We then implemented transfer learning and used VGG16 and Xception models to build models on both these datasets.

Here we conclude the final accuracy, precision, recall and f1-score for each of these models for both the datasets and draw a comparison of each of these models in each dataset.

After evaluating these values, we conclude that an ensemble classifier of our custom CNN model and VGG16 predicts an overall good result. While the Xception model does not work for our dataset. A good precision value of 92% is observed form the VGG16 model in predicting Pneumonia cases while we get a better precision value of 92% for the custom neural network in predicting normal cases.

We choose precision as the final metric of evaluation because it in medical proceedings, False positive values are costly because a patient may not have Pneumonia and the model may predict a positive value. Thus, a higher precision value is better for our model since it eliminates false positive values.

Here, we show the results:

Balanced Dataset			
	Precision (%)	Recall (%)	F1-score (%)
	Custom		
Normal Class	92	79	85
Pneumonia Class	89	96	92
Accuracy (%)	90		
	VGG16		
Normal Class	91	86	89
Pneumonia Class	92	95	94
Accuracy (%)	92		
	Xception		
Normal Class	39	97	55
Pneumonia Class	81	8	14
Accuracy (%)	41		

Imbalanced Dataset			
	Precision (%)	Recall (%)	F1-score (%)
	Custom		
Normal Class	92	79	85
Pneumonia Class	89	96	92
Accuracy (%)	90		
	VGG16		
Normal Class	76	97	85
Pneumonia Class	98	82	89
Accuracy (%)	87		
	Xception		
Normal Class	81	66	73
Pneumonia Class	82	91	86
Accuracy (%)	82		

Bibliography:

- Mooney, Paul. “Chest X-Ray Images (Pneumonia).” Kaggle, March 24, 2018. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia/metadata

- E. Ayan and H. M. Ünver, "Diagnosis of Pneumonia from Chest X-Ray Images Using Deep Learning," 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), Istanbul, Turkey, 2019, pp. 1-5. https://ieeexplore.ieee.org/document/8741582

- J. Ker, L. Wang, J. Rao and T. Lim, "Deep Learning Applications in Medical Image Analysis," in IEEE Access, vol. 6, pp. 9375-9389, 2018. https://ieeexplore.ieee.org/abstract/document/8241753

- Pneumonia Statistics 2019 [online]: https://www.cdc.gov/pneumonia/prevention.html.

- S.-H. Tsang Review: Xception—With Depthwise Separable Convolution Better Than Inception-v3 2018 [online]: https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568.

- D. A. Ragab, M. Sharkas, S. Marshall, and J. Ren, "Breast cancer detection using deep convolutional neural networks and support vector machines," PeerJ, vol. 7, p. e6201, 2019: https://peerj.com/articles/6201/

- S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images,"IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1240-1251, 2016: https://www.ncbi.nlm.nih.gov/pubmed/26960222

- W. H. Organization, "Standardization of interpretation of chest radiographs for the diagnosis of pneumonia in children," 2001: https://apps.who.int/iris/handle/10665/66956