Chest X-ray classification with CNN

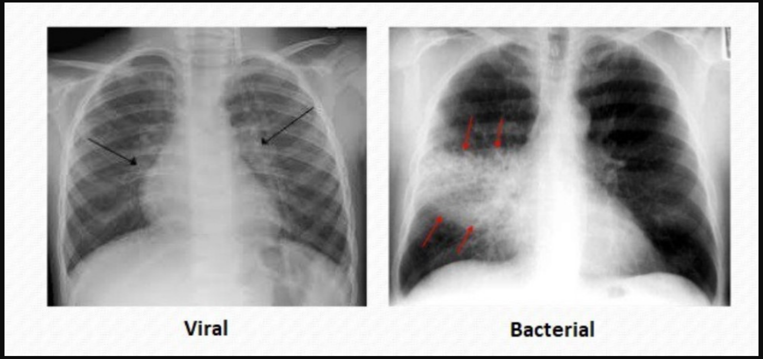
PHY 555, Deep Learning

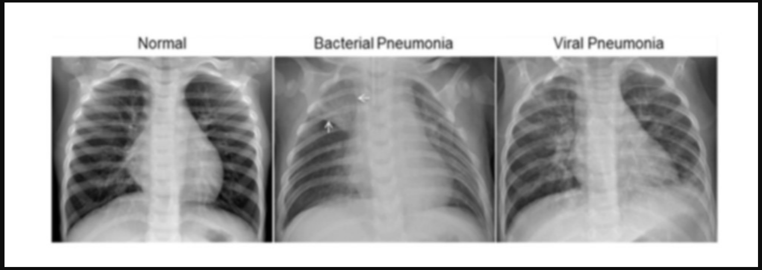
Mukesh Aryal

2022-04-12

Context:

- Dataset contains three types of x-ray images: normal, people with bacterial pneumonia, people with viral pneumonia.
- Images differ from each other on the basis on abnormal opacification in the image.
- Normal case: It includes x-ray images showing no signs of abnormal opacification.
- Bacterial case: Images with this case exhibits a focal lobar consolidation.
- Viral case: Images depict a more diffuse interstitial pattern in both lungs.

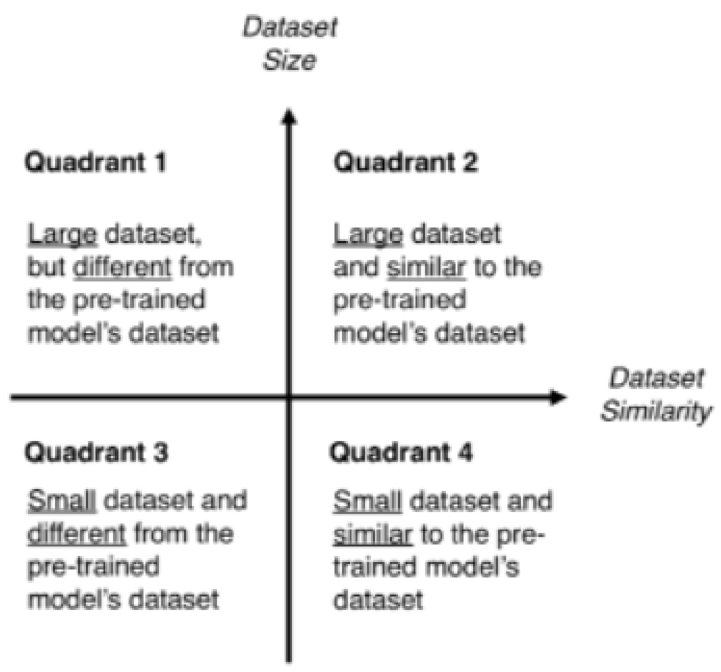


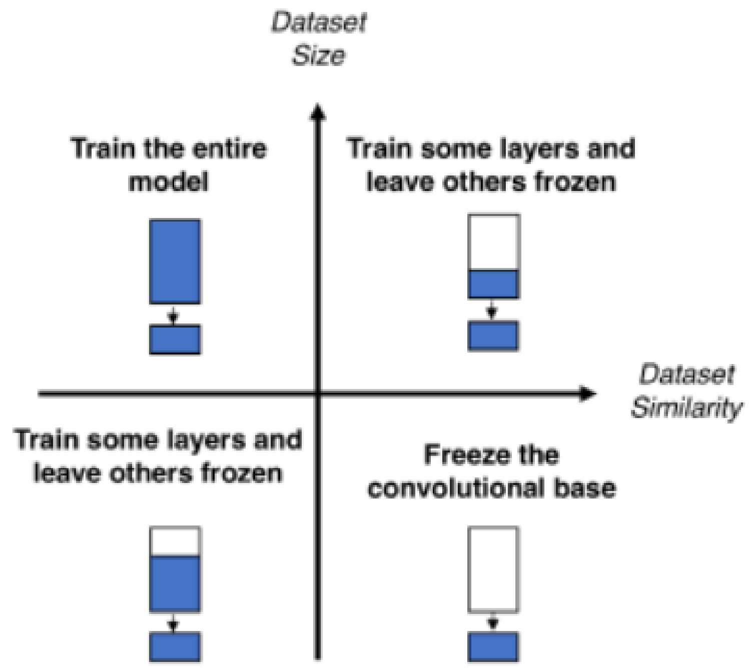


Objective and Approach

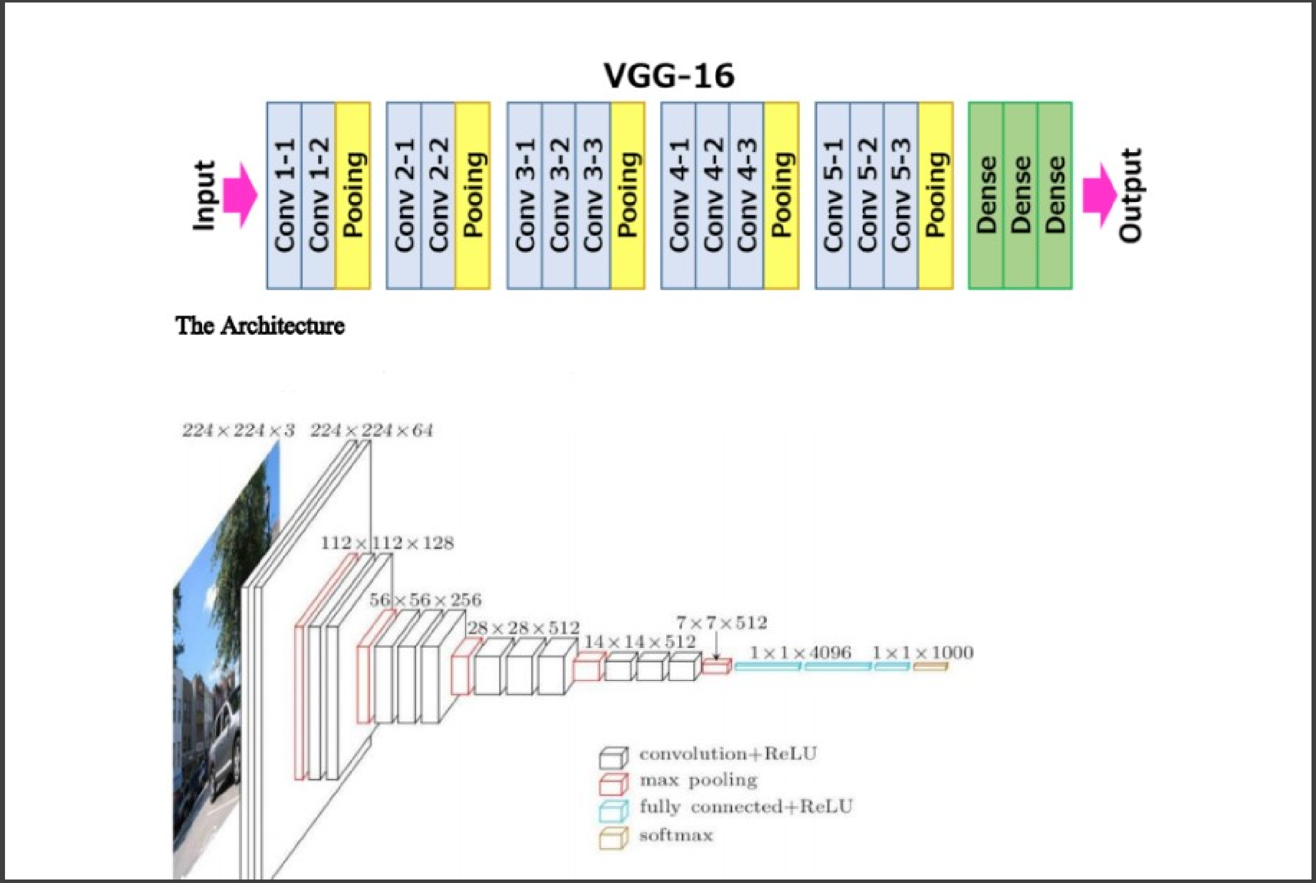
- To study feasibility, strength and limitations of transfer learning.
- Implementation of a simple CNN architecture and using it for classification tasks.
- Transfer learning using pre-trained VGG16 model for classification tasks.
- Comparison between two approaches and their performance.
- Classification task included:
    - Classification of images into two categories: Normal and Pneumonia cases.
    - Classification of images into three categories: Bacterial, Normal and Viral cases.

Transfer Learning:


Dataset Size
Dataset Similarity
Quadrant 1
Large dataset, but different from the pre-trained model's dataset
Quadrant 2
Large dataset and similar to the pre-trained model's dataset
Quadrant 3
Small dataset and different from the pre-trained model's dataset
Quadrant 4
Small dataset and similar to the pre-trained model's dataset

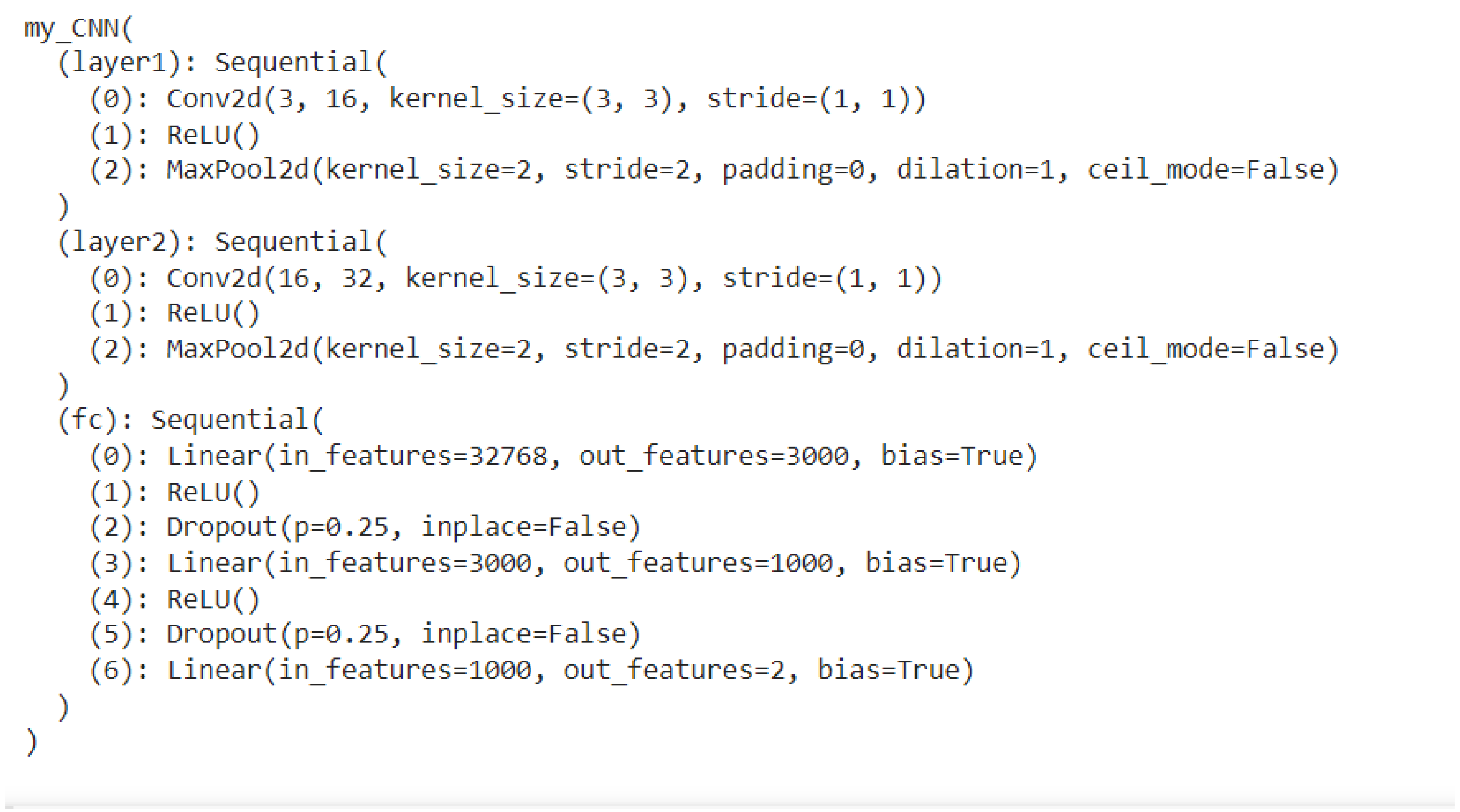

Dataset Size
Dataset Similarity
Train the entire model
Train some layers and leave others frozen
Train some layers and leave others frozen
Freeze the convolutional base

VGG16 Architecture:



- First approach was to test the pretrained network by freezing the convolutional block (preventing back propagation) and modifying the classifier to output the number of the classes required.
- Second approach was to partially freeze the convolutional block while allowing few of the last layers of the convolutional block to work better with the model.

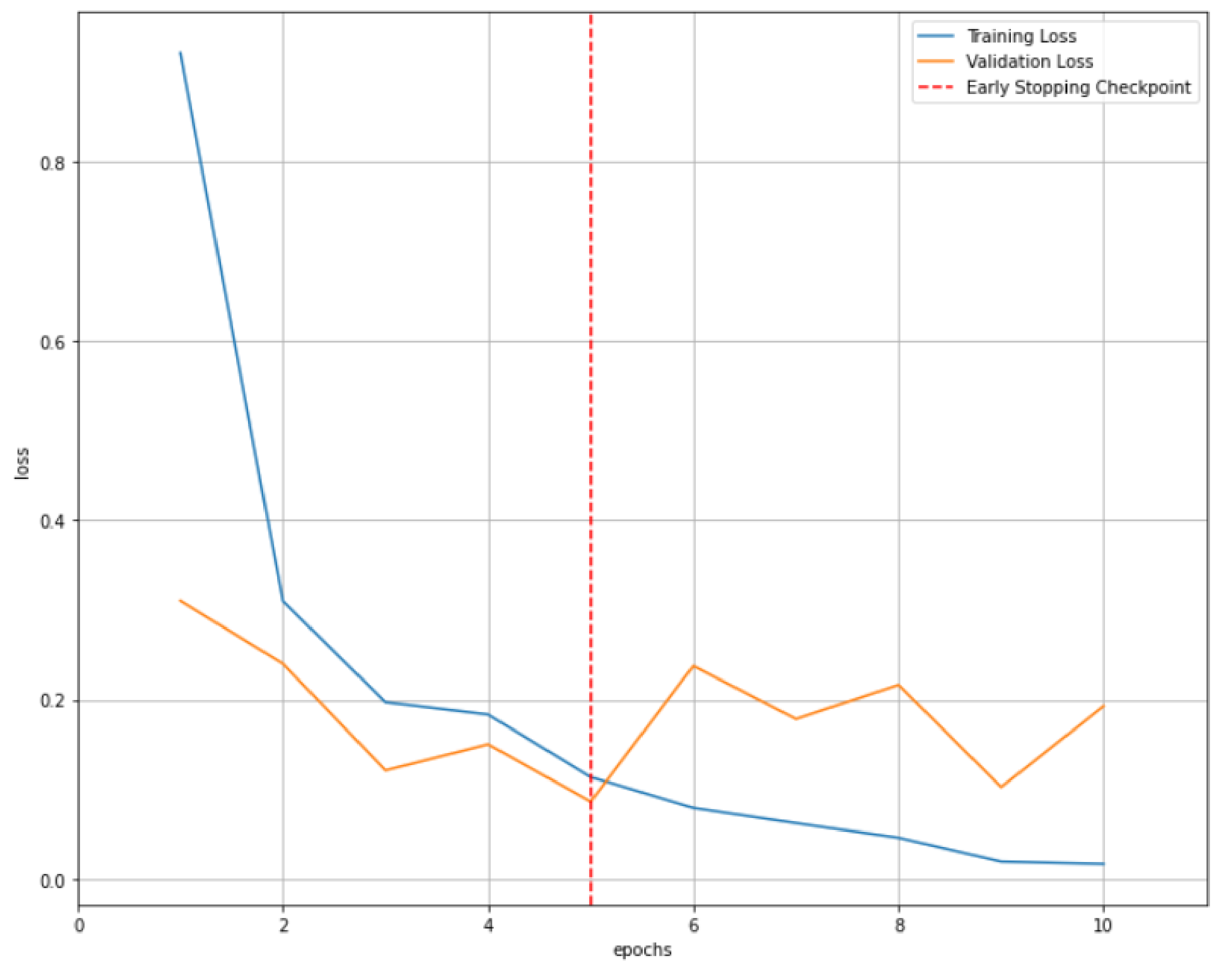
Simple CNN model:



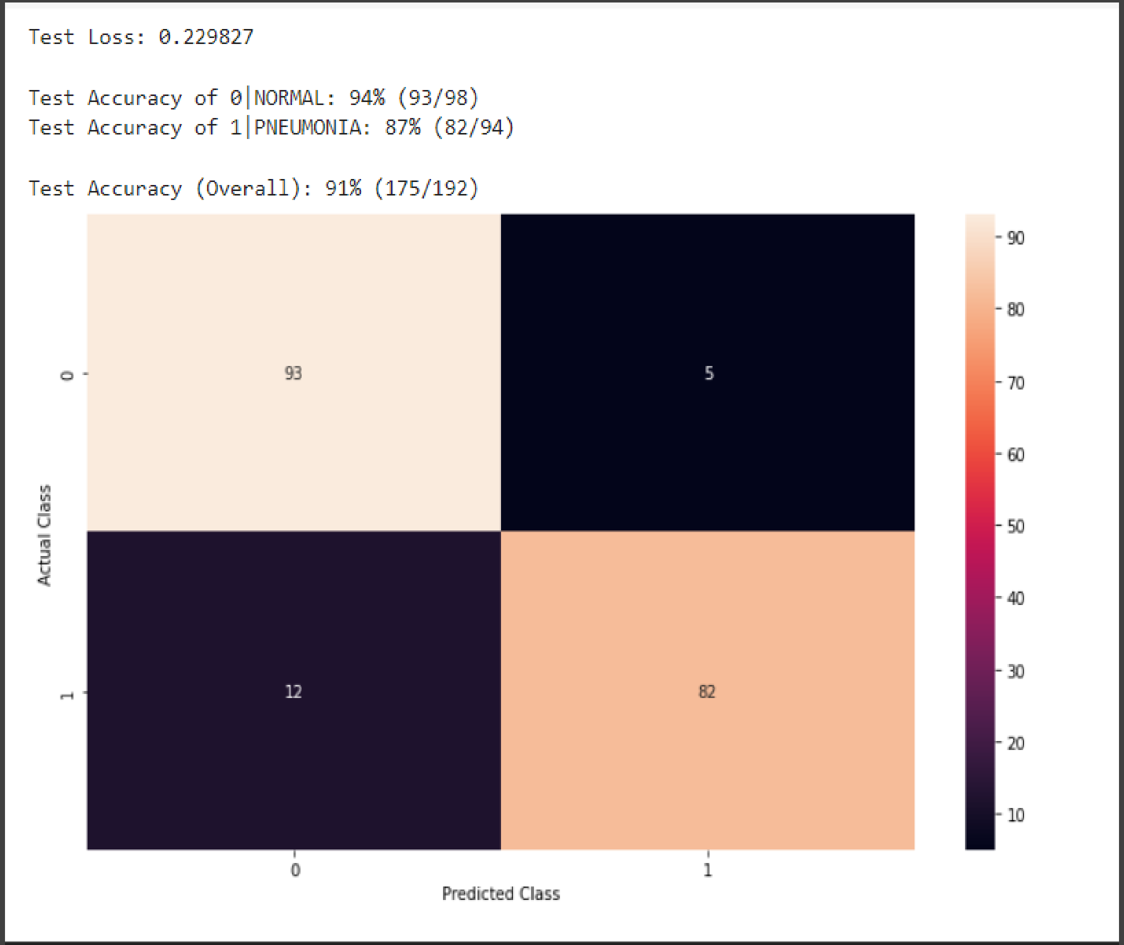
```
my_CNN(
  (layer1): Sequential(
    (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (layer2): Sequential(
    (0): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fc): Sequential(
    (0): Linear(in_features=32768, out_features=3000, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.25, inplace=False)
    (3): Linear(in_features=3000, out_features=1000, bias=True)
    (4): ReLU()
    (5): Dropout(p=0.25, inplace=False)
    (6): Linear(in_features=1000, out_features=2, bias=True)
  )
)
```

Early stopping:

- Use of early stopping to monitor overfitting.
- The image shows the result while classifying the images in two categories: Normal and Pneumonia using simple CNN model.

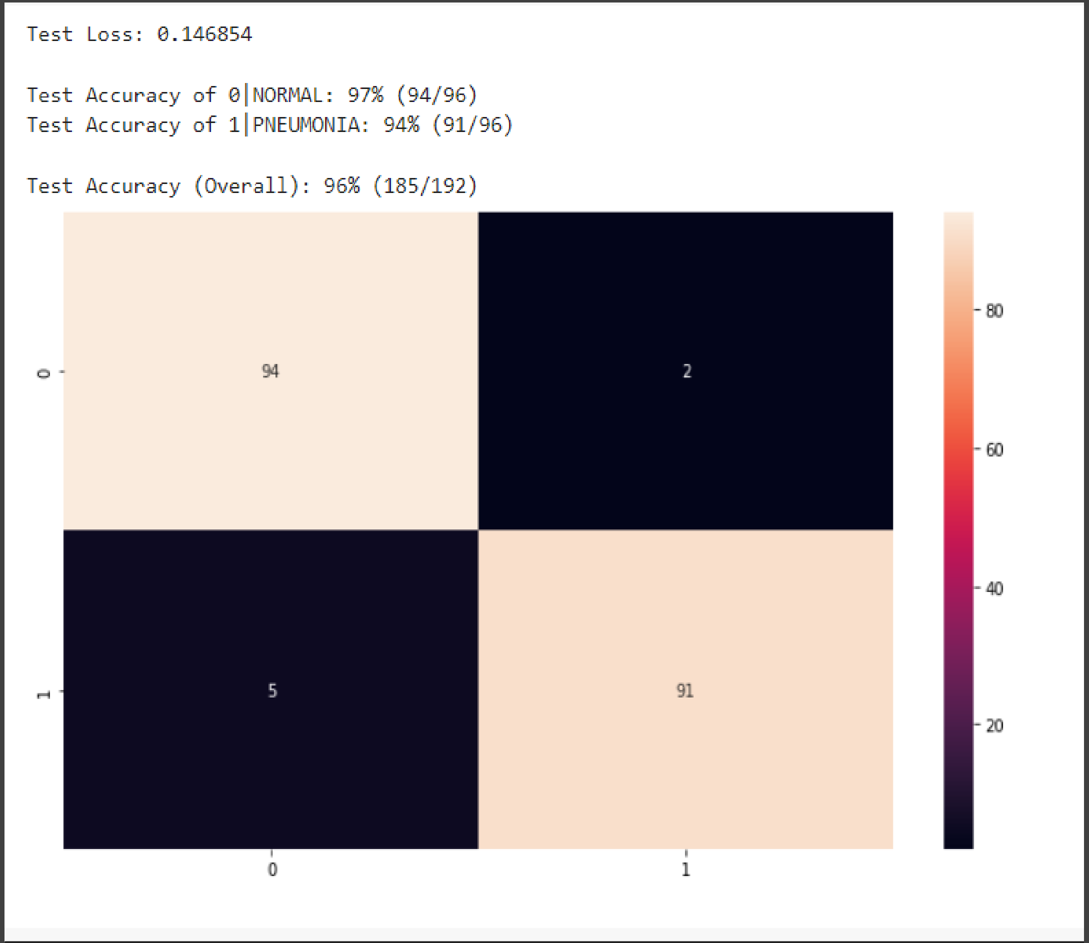


Results and comparison (Two classes):



```
Test Loss: 0.229827

Test Accuracy of 0|NORMAL: 94% (93/98)
Test Accuracy of 1|PNEUMONIA: 87% (82/94)

Test Accuracy (Overall): 91% (175/192)
```

VGG16

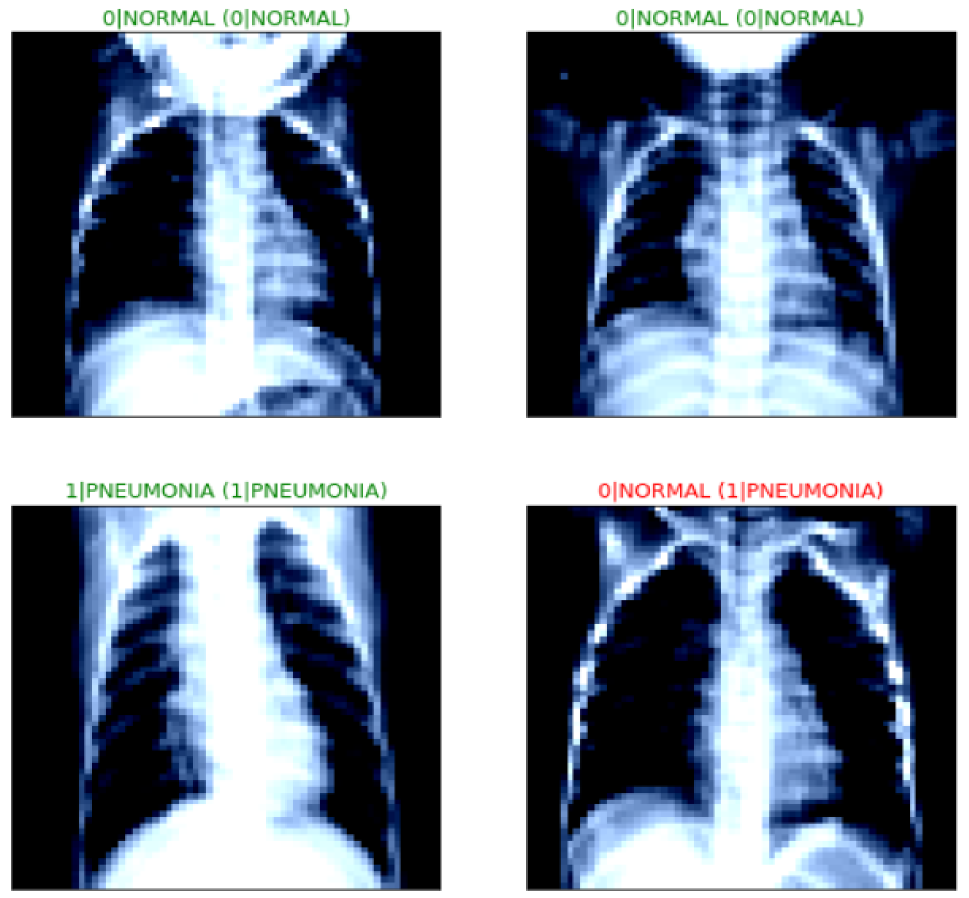


```
Test Loss: 0.146854

Test Accuracy of 0|NORMAL: 97% (94/96)
Test Accuracy of 1|PNEUMONIA: 94% (91/96)

Test Accuracy (Overall): 96% (185/192)
```

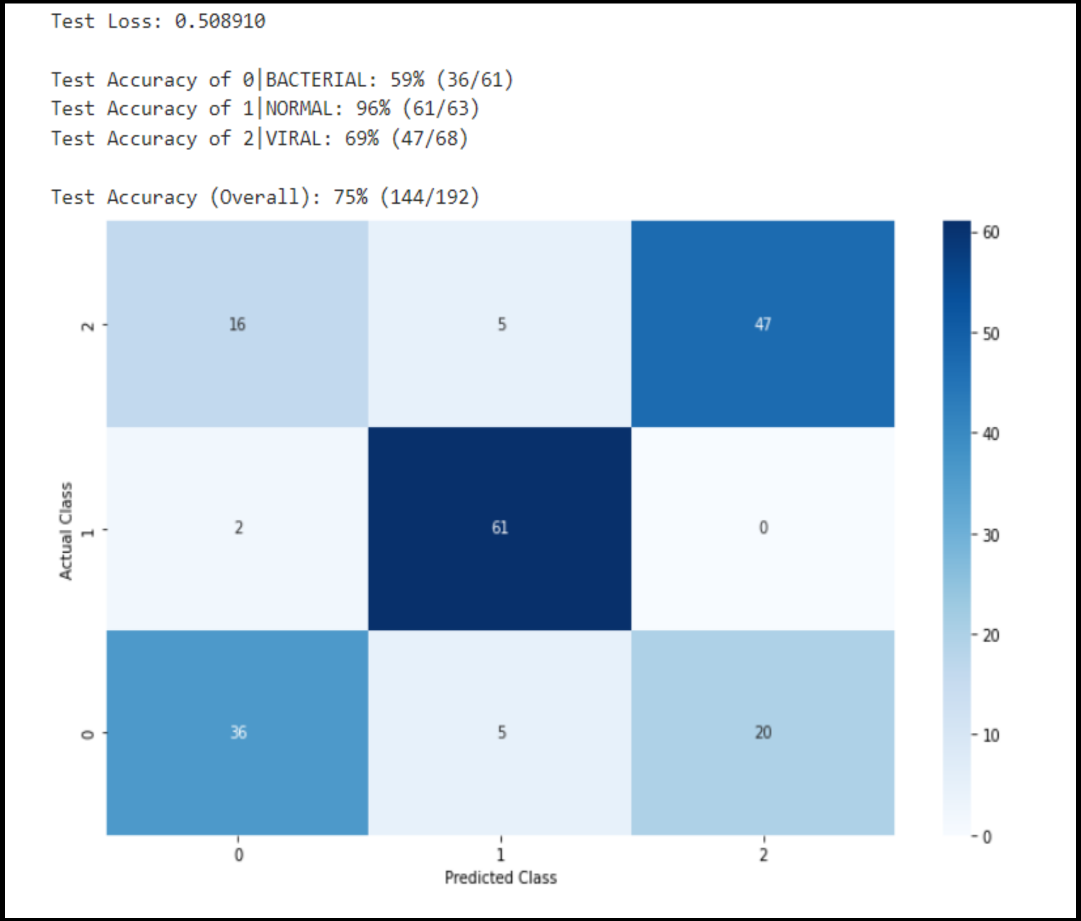
Simple CNN

Example Predictions

Prediction from VGG16. Ground truths are represented inside the brackets.

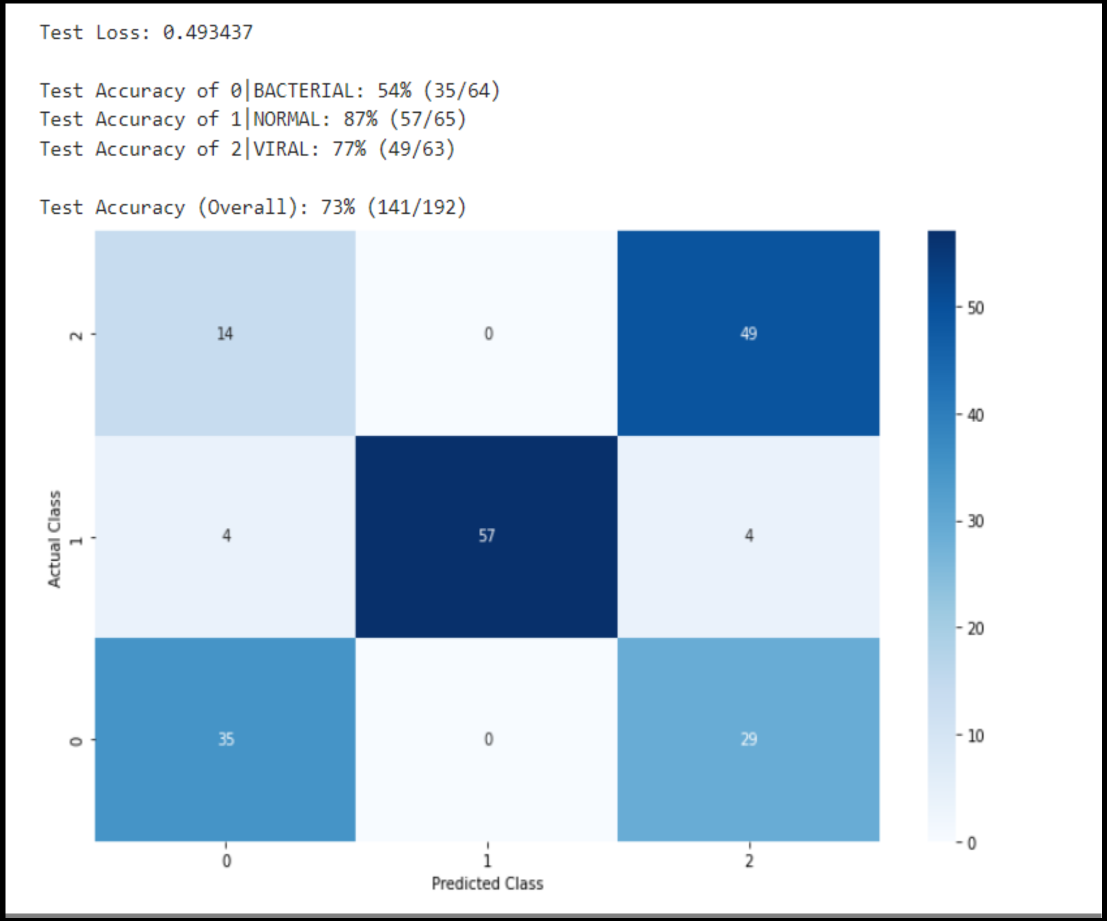


Results and comparison (Three classes):



```
Test Loss: 0.508910

Test Accuracy of 0|BACTERIAL: 59% (36/61)
Test Accuracy of 1|NORMAL: 96% (61/63)
Test Accuracy of 2|VIRAL: 69% (47/68)

Test Accuracy (Overall): 75% (144/192)
```

VGG16

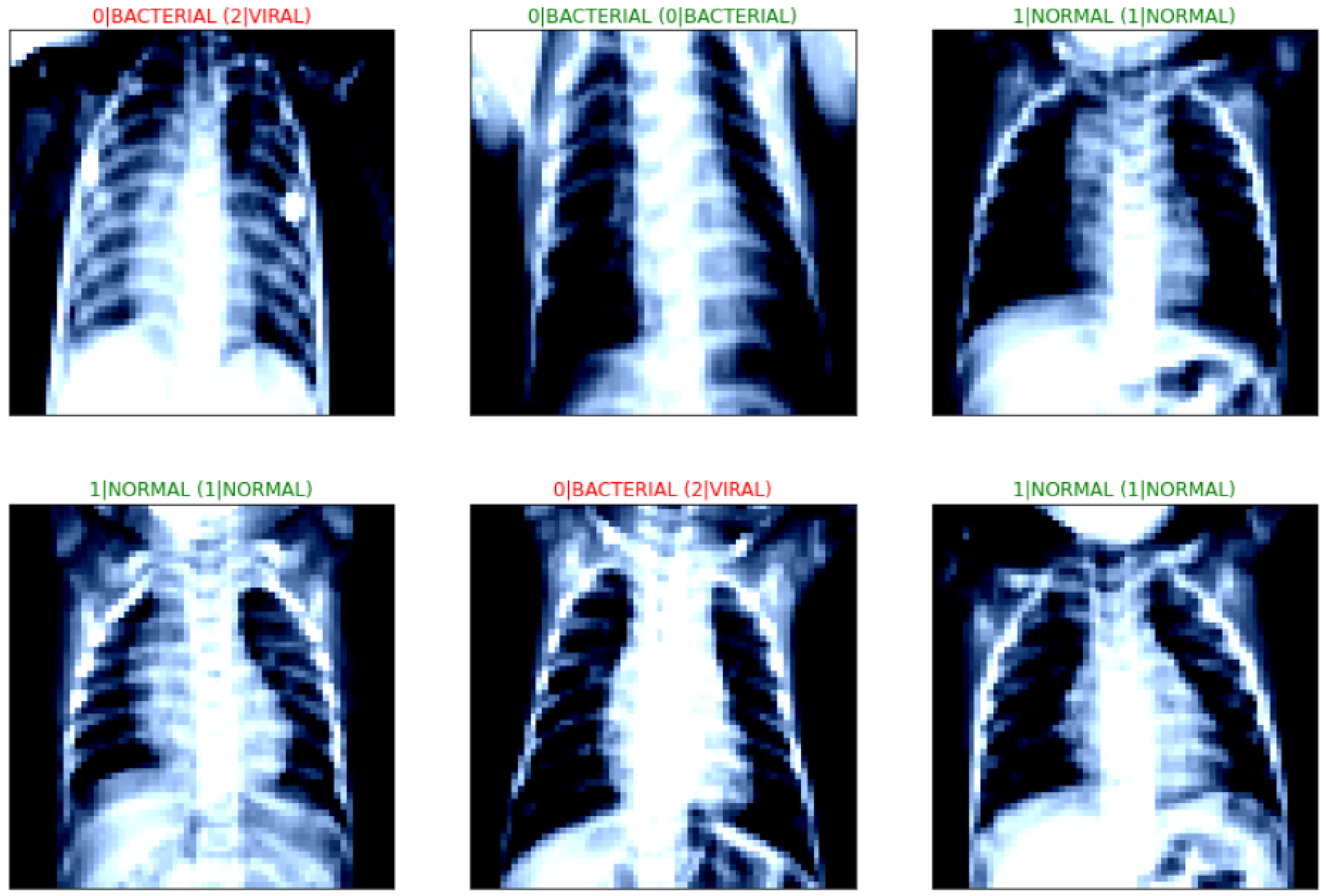


```
Test Loss: 0.493437

Test Accuracy of 0|BACTERIAL: 54% (35/64)
Test Accuracy of 1|NORMAL: 87% (57/65)
Test Accuracy of 2|VIRAL: 77% (49/63)

Test Accuracy (Overall): 73% (141/192)
```

Simple CNN

Example Predictions

- Prediction from VGG. Ground truths are represented inside the brackets. Network is mostly misclassifying Viral and Bacterial cases.



Remarks:

CNN, although more efficient than fully connected networks, are still computationally expensive. If the problem can be solved by easier approach, then avoiding its use can save time.

Using pretrained models can be most helpful in situations where you have dataset similar to the ones that the network is trained on.

If the dataset is not very similar, then changing the classifier of the pre-trained model alone may not be sufficient. Training few end layers or most layers of Convolutional blocks may increase the performance of the transfer model.

Despite the limitations of transfer learning, they are very easy to work with instead of building your own network from scratch.