CS440 Introduction to AI

Professor Sarah Adel Bargal

December 6, 2021

Class Challenge Report: Image Classification of COVID-19 X-rays

Lim, Chiara

Qasim, Junaid

Email: cclim@bu.edu, jqasim1@bu.edu

Task 1 - Binary Classification (Normal vs. COVID-19 X-rays)

Architecture:

For Task 1, we used a CNN model, VGG16, that used pre-trained weights obtained by training on ImageNet. A VGG16 model consists of the 16 layers shown in figure 1.1.2. This was followed by 1 flatten layer (named “flatten”), 1 dense layer (named “dense_feature”), 1 dropout layer (named “dropout”), and lastly, 1 dense layer (named “dense”). The first dense layer, “dense_feature,” was used as a feature extraction layer with a RELU activation function, and had 256 nodes. The “dropout” layer had a dropout rate of 0.25 and 256 nodes as well. Then, the second and final dense layer used the sigmoid activation function. This layer had 1 unit for classification since there are only 2 classes of images - normal and COVID-19. The model contained a total of 16 + 3 = 19 layers (Fig. 1.1.1).

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Functional)           (None, 7, 7, 512)         14714688
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense_feature (Dense)        (None, 256)               6422784
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 257
=================================================================
Total params: 21,137,729
Trainable params: 6,423,041
Non-trainable params: 14,714,688
_________________________________________________________________
```

Figure 1.1.1 VGG16 Model Layers

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 64)      1792
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 224, 224, 64)      36928
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 112, 112, 64)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 112, 112, 128)     73856
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 112, 112, 128)     147584
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 56, 56, 256)       295168
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 56, 56, 256)       590080
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 56, 56, 256)       590080
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 256)       0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 28, 28, 512)       1180160
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 28, 28, 512)       2359808
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 28, 28, 512)       2359808
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 512)       0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense (Dense)                (None, 4096)              102764544
_________________________________________________________________
dropout (Dropout)            (None, 4096)              0
_________________________________________________________________
dense_1 (Dense)              (None, 4096)              16781312
_________________________________________________________________
dropout_1 (Dropout)          (None, 4096)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1000)              4097000
=================================================================
Total params: 138,357,544
Trainable params: 138,357,544
Non-trainable params: 0
```

Figure 1.1.2 VGG16 Model Layers

The following table consists of hyperparameters we used to fine-tune our VGG16 model:

Hyperparameter	Name
Optimizer	Adam
Loss function	Binary cross-entropy
Parameters	• Total parameters: 21,137,729 • Trainable parameters: 6,423,041 • Non-trainable parameters: 14,714,688
Regularization	None
Learning rate	0.0001
Batch size	10
Number of Epochs	40

Figure 1.2 VGG16 Model Parameters

Model Accuracy and Loss:
After training was completed, on average, our model produced a training accuracy of 0.95 and a training loss of 0.15 (Fig. 1.3-1.4). Our model also produced a test accuracy of 1.0 and a test loss of approximately 0.09 (Fig. 1.5).

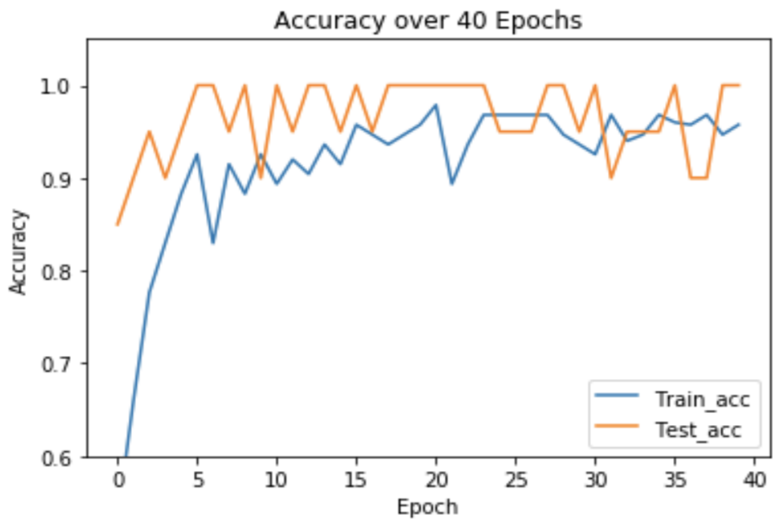

Figure 1.3 VGG16 Model Accuracy

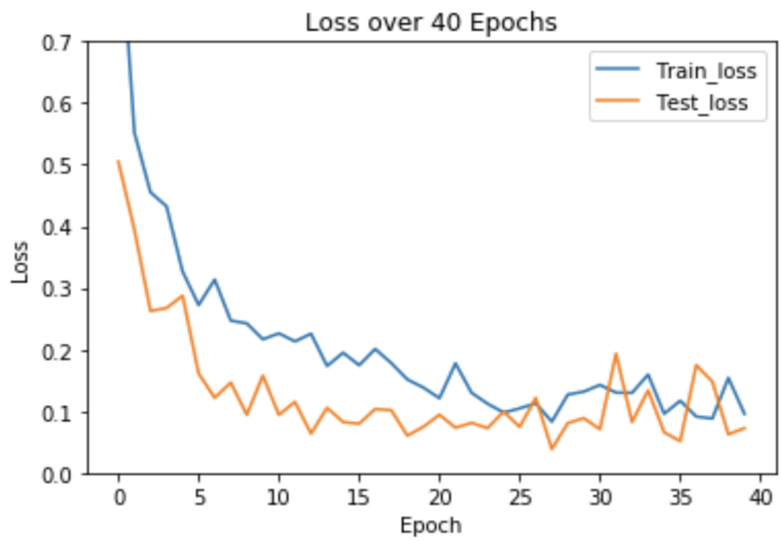

Figure 1.4 VGG15 Model Loss

```
Found 18 images belonging to 2 classes.
18/18 [==============================] - 4s 225ms/step - loss: 0.0934 - accuracy: 1.0000
Test loss:  0.09344383329153061
Test accuracy:  1.0
```

Figure 1.5 VGG16 Model Testing Accuracy & Loss

T-SNE Visualization:

We visualized features of the training data by extracting the "dense feature" layer for the t-SNE plot. By observing the t-SNE graph, we concluded that the model performed relatively well, as seen by the 2 distinct clusters representing COVID-19 (blue) and normal (orange) data points (Fig. 1.6).

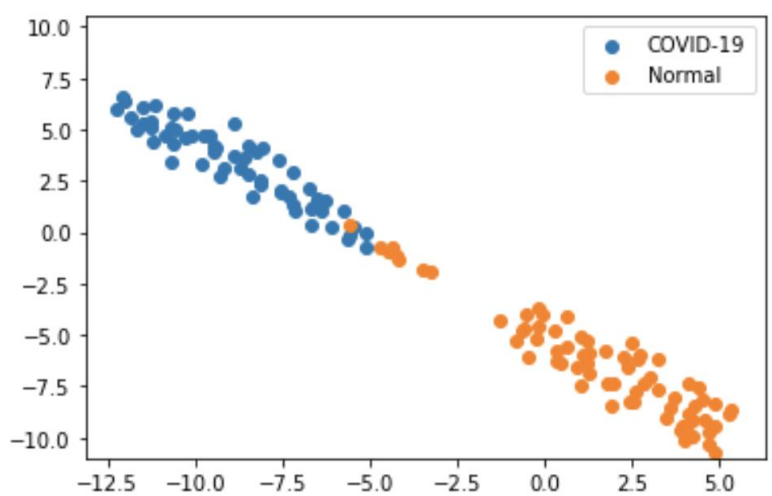

Figure 1.6 VGG16 t-SNE Visual

Task 2 - Multi Classification (Normal, COVID-19, Pneumonia-Bacterial, and Pneumonia-Viral)

Architecture:

For Task 2, we compared a VGG16 and ResNet50, both of which used pre-trained weights obtained by training on ImageNet. The VGG16 model was followed by 1 flatten layer (named “flatten”), 1 dense layer (named “dense_feature”), 1 dropout layer (named “dropout_1”), and lastly, 1 dense layer (named “dense”). The first dense layer, “dense_feature,” was used as a feature extraction layer, with a RELU activation function, and had 256 nodes. The “dropout_1” layer had a dropout rate of 0.5. The model outputted to the second and last “dense” layer which used the softmax activation function. This layer had 4 units for classification since there were 4 classes of images - normal, COVID-19, Pneumonia-Bacterial, and Pneumonia-Viral. The model contained a total of 16 + 4 = 20 layers (Fig 1.1.2, 2.1.1).

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Functional)           (None, 7, 7, 512)         14714688
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense_feature (Dense)        (None, 256)               6422784
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 4)                 1028
=================================================================
Total params: 21,138,500
Trainable params: 6,423,812
Non-trainable params: 14,714,688
_________________________________________________________________
```

Figure 2.1.1 VGG16 Model Layers

A ResNet50 model consists of 50 layers. The ResNet50 model was followed by 1 2D convolutional layer (named “conv2d_8”), 1 flatten layer (named “flatten”), 1 dense layer (named “dense_feature”), 1 dropout layer (named “dropout”), and lastly, 1 dense layer (named “dense”). The “conv2d_8” layer convoluted the output of the ResNet layer to 1024 nodes with a kernel size of (1,1) and padding. The first dense layer, “dense_feature”, was used as a feature extraction layer, with 256 nodes and a RELU activation function. The “dropout” layer had a dropout rate of 0.25. The model outputted to the second and last “dense” layer, which used the softmax activation function. This layer had 4 units for classification since there were 4 classes of images - normal, COVID-19, Pneumonia-Bacterial, and Pneumonia-Viral. The model contained a total of 50 + 5 = 55 layers (Fig 2.1.2).

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Functional)        (None, 7, 7, 2048)        23587712
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 7, 7, 1024)        2098176
_________________________________________________________________
flatten (Flatten)            (None, 50176)             0
_________________________________________________________________
dense_feature (Dense)        (None, 128)               6422656
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 4)                 516
=================================================================
Total params: 32,109,060
Trainable params: 32,055,940
Non-trainable params: 53,120
_________________________________________________________________
```

Figure 2.1.2 ResNet50 Model Layers

The following table consists of hyperparameters we used to fine-tune our VGG16 model:

Hyperparameter	Name
Optimizer	Adam
Loss function	Categorical cross-entropy
Parameters	• Total parameters: 21,138,500 • Trainable parameters: 6,423,812 • Non-trainable parameters: 14,714,688
Regularization	None
Learning rate	0.0001
Batch size	10
Number of Epochs	100

Figure 2.2.1 VGG16 Model Parameters

The following table consists of hyperparameters we used to fine-tune our ResNet50 model:

Hyperparameter	Name
Optimizer	Adam
Loss function	Categorical cross-entropy
Parameters	• Total parameters: 32,109,060 • Trainable parameters: 32,055,940 • Non-trainable parameters: 53,120
Regularization	None
Learning rate	0.00001
Batch size	10
Number of Epochs	100

Figure 2.2.2 ResNet50 Model Parameters

Model Accuracy and Loss:
After training was completed, the VGG16 model produced a test accuracy of approximately 0.64 and a test loss of approximately 0.76 (Fig. 2.3.1-2.3.3).

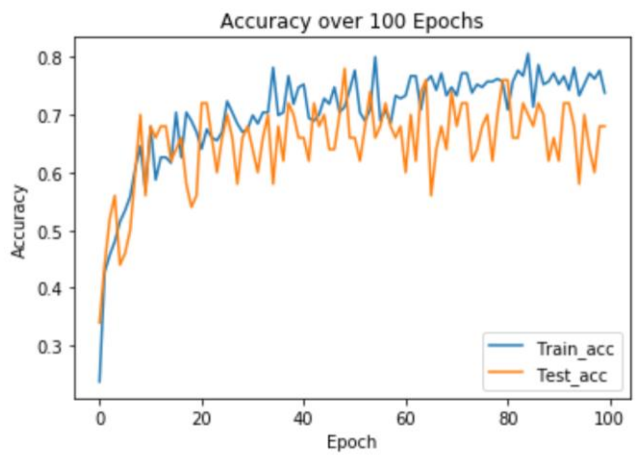

Figure 2.3.1 VGG16 Model Accuracy

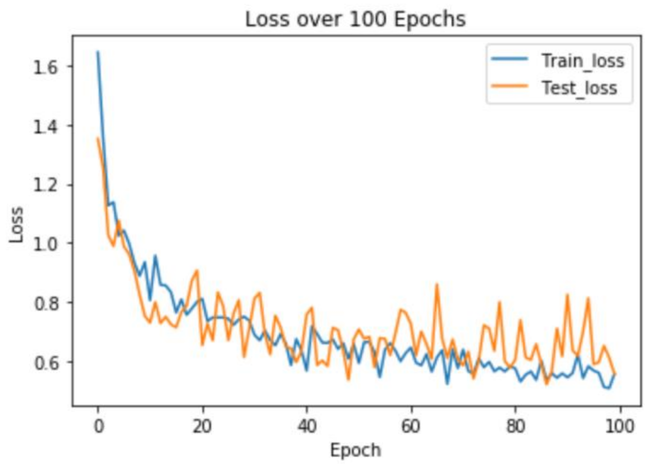

Figure 2.3.2 VGG16 Model Loss

```
Found 36 images belonging to 4 classes.
36/36 [==============================] - 13s 347ms/step - loss: 0.7578
- accuracy: 0.6389
Test loss: 0.7577974796295166
Test accuracy: 0.6388888955116272
```

Figure 2.3.3 VGG16 Test Accuracy and Test Loss

After training was completed, the ResNet50 model produced a test accuracy of approximately 0.69 and a test loss of approximately 0.86 (Fig. 2.4.1-2.4.3).

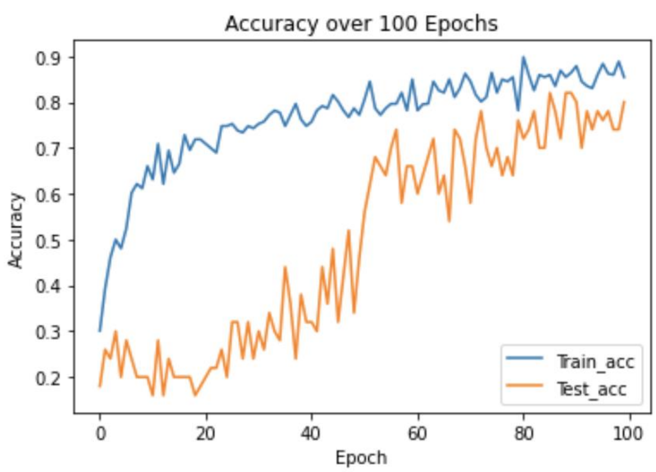

Figure 2.4.1 ResNet50 Model Accuracy

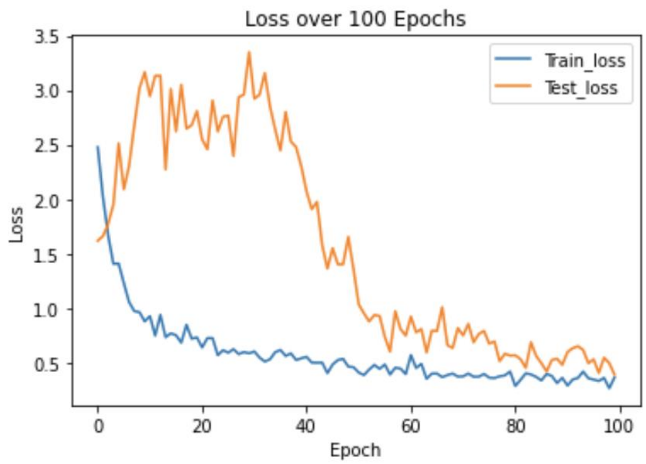

Figure 2.4.2 ResNet50 Model Loss

```
Found 36 images belonging to 4 classes.
36/36 [==============================] - 3s 90ms/step - loss: 0.8564 - accuracy: 0.6944
Test loss: 0.8563815951347351
Test accuracy: 0.6944444179534912
```

Figure 2.4.3 ResNet50 Model Test Accuracy & Loss

T-SNE Visualization:

For the VGG16 model, we visualized features of the training data by extracting the “dense feature” layer for the t-SNE plot. By observing the t-SNE graph, we can conclude that the model performed relatively decently, with 2 distinct clusters for COVID-19 and normal data points, but undistinguishable clusters for pneumonia-bacteria and pneumonia-viral (Fig. 2.5.1).

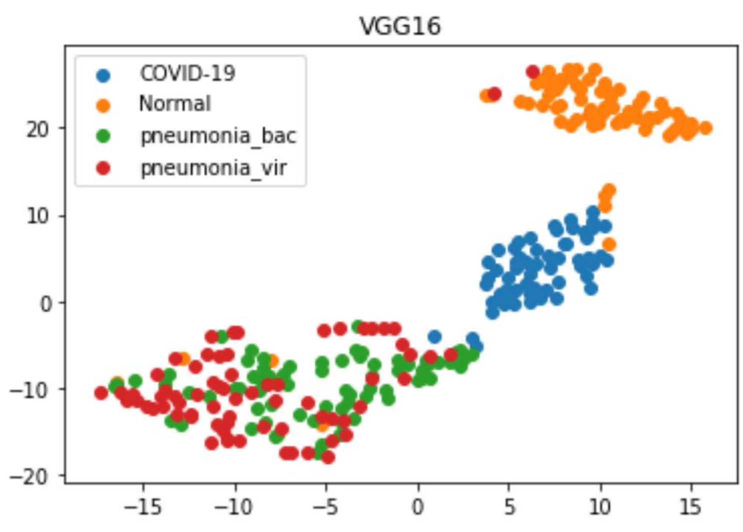

Figure 2.5.1 VGG16 t-SNE Visual

For the ResNet50 model, we visualized features of the training data by extracting the "dense feature" layer for the t-SNE plot. By observing the t-SNE graph, we can conclude that the model performed relatively decently, with 2 distinct clusters for COVID-19 and normal data points, but undistinguishable clusters for pneumonia-bacteria and pneumonia-viral (Fig. 2.5.2).

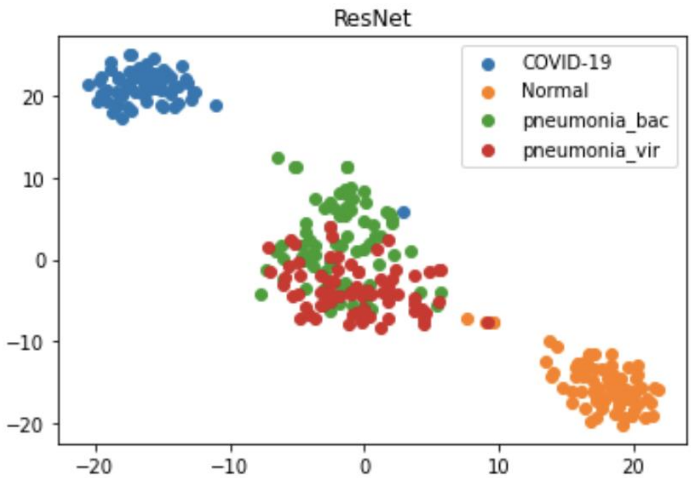

Figure 2.5.2 ResNet50 t-SNE Visual

Comparison of Performance:

For Task 2, we used the following 2 models: VGG16 and ResNet50. Comparing the two models' hyperparameters, the ResNet50 model had 32,109,060 total parameters - 32,055,940 (trainable) and 53,120 (non-trainable), whilst the VGG16 model had 21,138,500 total parameters - 6,423,812 (trainable) and 14,714,688 (non-trainable). As for the rest of the parameters, we kept them the same for both models as seen in figure 2.6.1.

Hyperparameter	ResNet50	VGG16
Optimizer	Adam	Adam
Loss function	Categorical cross-entropy	Categorical cross-entropy
Parameters	• Total parameters: 32,109,060 • Trainable parameters: 32,055,940 • Non-trainable parameters: 53,120	• Total parameters: 21,138,500 • Trainable parameters: 6,423,812 • Non-trainable parameters: 14,714,688
Regularization	None	None
Learning rate	**0.0001**	**0.0001**

Batch size	10	10
Number of Epochs	100	100

Figure 2.6.1

Model Analysis:

For a successful model, a higher test accuracy and a lower test loss are desirable. Furthermore, the t-SNE visualization produced by the model should correctly classify the 4 classes - COVID-19, normal, pneumonia-bacteria and pneumonia-viral. This is seen through the clear separation of data points into 4 distinct clusters.

In terms of test accuracy, the ResNet50 model performed better than VGG16 with test accuracies of 0.69 and 0.64 respectively. As for the test loss, the ResNnet50 model also performed better than VGG16 with test losses of 0.86 and 0.89 respectively. In terms of the t-SNE visualization, the ResNet50 model appears to have a clearer distinction between the COVID-19 and normal X-rays data points. However, both models failed to successfully separate the 2 pneumonia clusters (seen by the mix of red and green data points). Although the test and training accuracies and losses for the VGG16 model were more consistent with each other, the ResNet50 model had the best performance overall.

Upon reflection, introducing overfitting-prevention techniques such as regularization and early-stopping into training could have improved the overall performance of both models. Despite adding dropout layers in both models, the test accuracies were relatively low and the test losses were relatively high. Therefore, replacing or even adding different regularization techniques might have reduced the test losses and in turn, increased test accuracies. Overall, with the given time and limited resources, we believe our model modestly showcased our ability to perform binary and multi-class classification.[1]

[1] We really tried. Thank you for reading our report, and have a wonderful winter break :)