CS542 Class Challenge Report:
Image Classification of COVID-19 X-rays

Yu Xia

U78899432

Prof. Sarah Adel Bargal

April 29, 2021

Contents

Introduction

In this class challenge we are classifying X-ray images of lungs. The data we will use has been collected by Adrian Xu, combining the Kaggle Chest X-ray dataset with the COVID-19 Chest X-ray dataset collected by Dr. Joseph Paul Cohen of the University of Montreal.

There are two tasks in total. In task 1, we need to train a deep neural network model to classify normal vs. COVID-19 X-rays. In the second task, we need to train a deep neural network model to classify an X-ray image into one of the following classes: normal, COVID-19, Pneumonia-Bacterial, and Pneumonia-Viral.

Task 1: Binary Classification

Architecture

I implemented a VGG16 model for this task:

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

The benefit of using a VGG16 model is that it's very effective with image classification. The architecture has 3 layers: a flatten layer, a fully connected layer with ReLu because it works with CNN and a fully connected function with Sigmoid.

Optimizer, Loss Function, Parameters and Regularization

Optimizer: Adam.

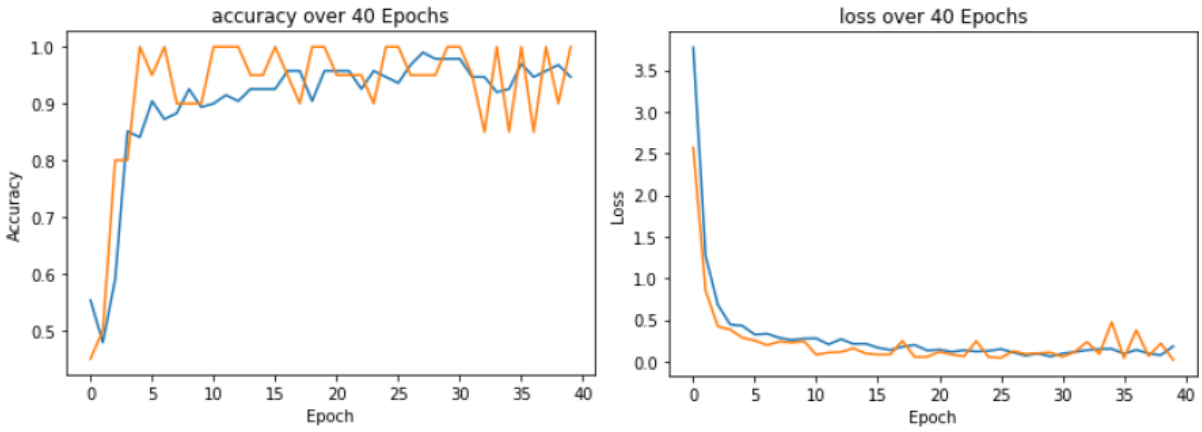
Loss function: tf.keras.losses.BinaryCrossentropy

Epochs: 40.

Batch size: 10.

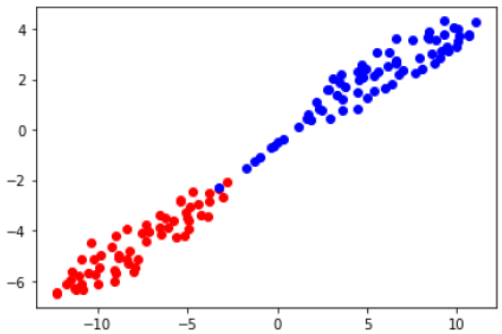
Accuracy and Loss

The blue lines are accuracy and loss, and the orange lines are validation accuracy and loss. In the accuracy plot, the validation accuracy and train accuracy are close with each other. It is the same case for the loss plot. We can safely assume the model is not overfitting.



t-SNE Visualizations

The red points are image of Covid X-ray and the blue points are normal X-rays. The points are mostly well divided into two group. It is a good classifier.



Task 2: Multi-Class Classification

Architecture

Two different architectures were implemented for this task.

The VGG16 architecture:

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Functional)           (None, 7, 7, 512)         14714688
_________________________________________________________________
flatten_7 (Flatten)          (None, 25088)             0
_________________________________________________________________
dropout_14 (Dropout)         (None, 25088)             0
_________________________________________________________________
dense_21 (Dense)             (None, 2048)              51382272
_________________________________________________________________
dropout_15 (Dropout)         (None, 2048)              0
_________________________________________________________________
feature_dense (Dense)        (None, 256)               524544
_________________________________________________________________
dense_22 (Dense)             (None, 4)                 1028
=================================================================
Total params: 66,622,532
Trainable params: 51,907,844
Non-trainable params: 14,714,688
_________________________________________________________________
```

The ResNet50 architecture:

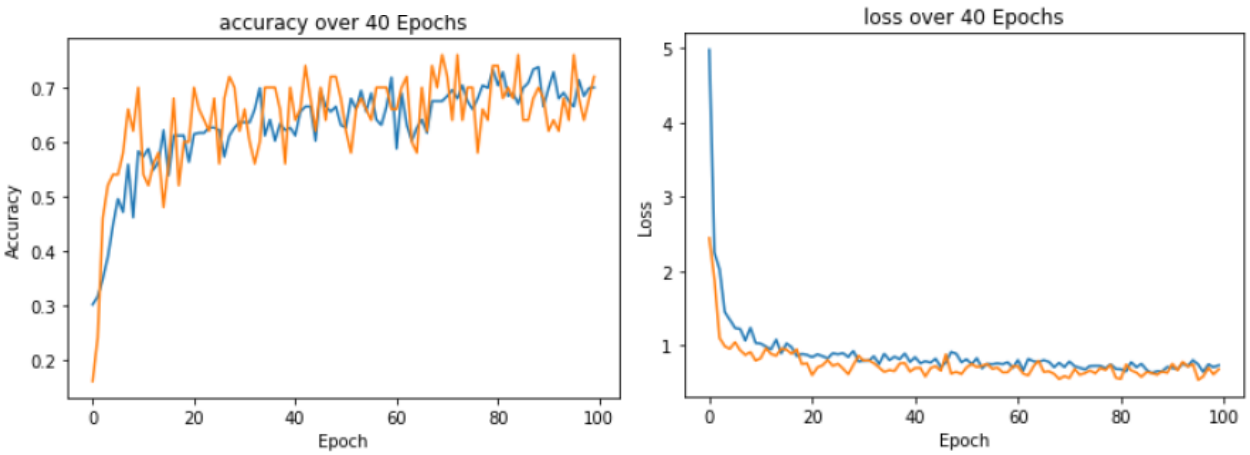
Optimizer, Loss Function, Parameters and Regularization

Optimizer: Adam.
Loss function: tf.keras.losses.CategoricalCrossentropy
Epochs: 100.
Regularization: Dropout. Two layers of dropout are employed to prevent overfitting. The two dropout layers are set to 0.5 and 0.4 after trying out a few values.

Batch size: 10.

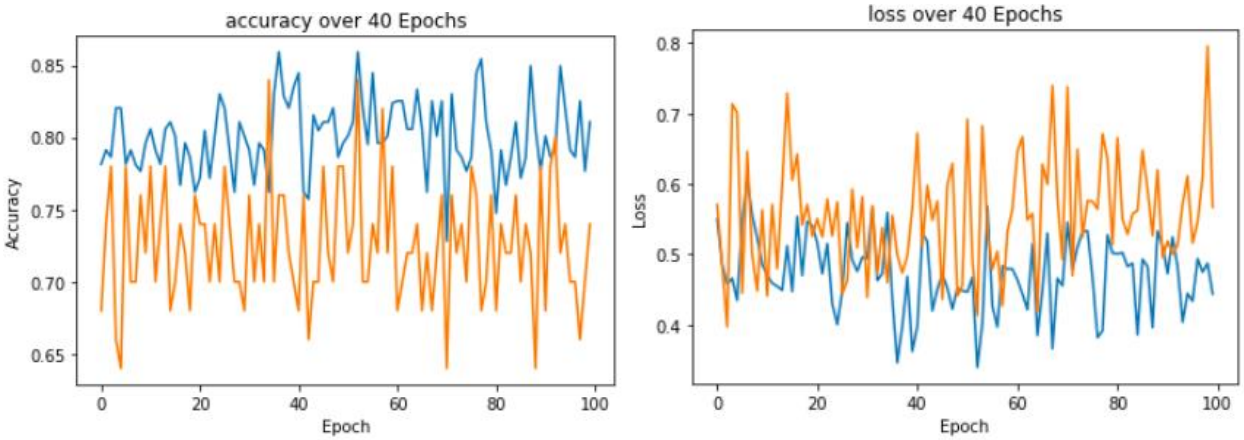
Accuracy and Loss

VGG16:



Both accuracy and plot converge after around 40 epochs. Validation accuracy and loss is close to train accuracy and loss, or slightly lower. It shows that the model is not overfitting.

ResNet50:

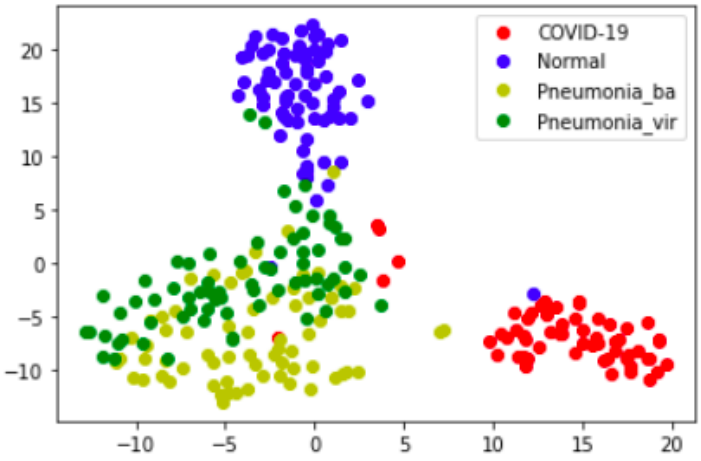


The validation accuracy and loss are very unstable. Validation accuracy is lower than train accuracy and validation loss is lower than train loss. This model could be overfitting.

t-SNE Visualizations

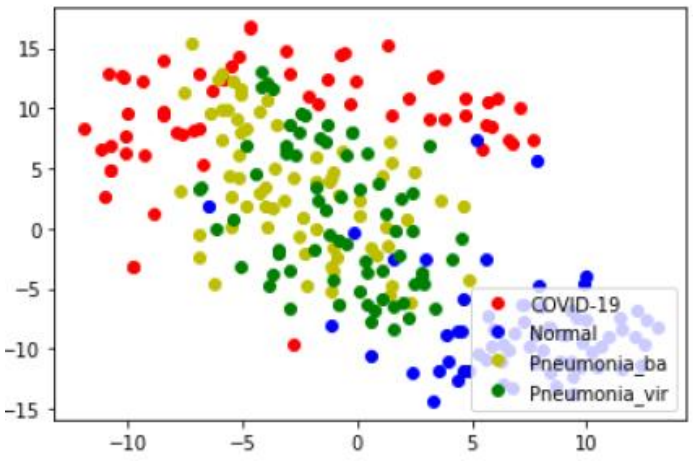
ResNet50:

The 4 classes of images are relatively well divided into 3 groups. The green and yellow points are in a same group, which proves they have similar features.



VGG16:

The 4 classes are not classified well enough in the plot. They are roughly in 3 groups. The red points and the blue points are distinguished in 2 corners of the plot. The green and yellow points have similar features and are mixed together in the middle of the plot.



ResNet50 and VGG16 Comparison

VGG16:

```
Found 36 images belonging to 4 classes.
36
36/36 [==============================] - 2s 66ms/step - loss: 0.7948 -
accuracy: 0.7500
Test loss: 0.7948353290557861
Test accuracy: 0.75
```

ResNet50:

```
Found 36 images belonging to 4 classes.
36
36/36 [==============================] - 1s 30ms/step - loss: 0.7694 -
accuracy: 0.6944
Test loss: 0.7693760991096497
Test accuracy: 0.6944444179534912
```

The test accuracy of the two models does not vary huge from each other. The VGG16 model has a higher accuracy by about 5 percent.

The ResNet50 model becomes overfitting and VGG16 did not. This shows that ResNet50 is likely more complex than VGG16 and so it's easier for it to become over fitting.

The t-SNE plot of the two model varies from each other. ResNet50 has more clearly distinguished clusters than VGG16.

Challenge