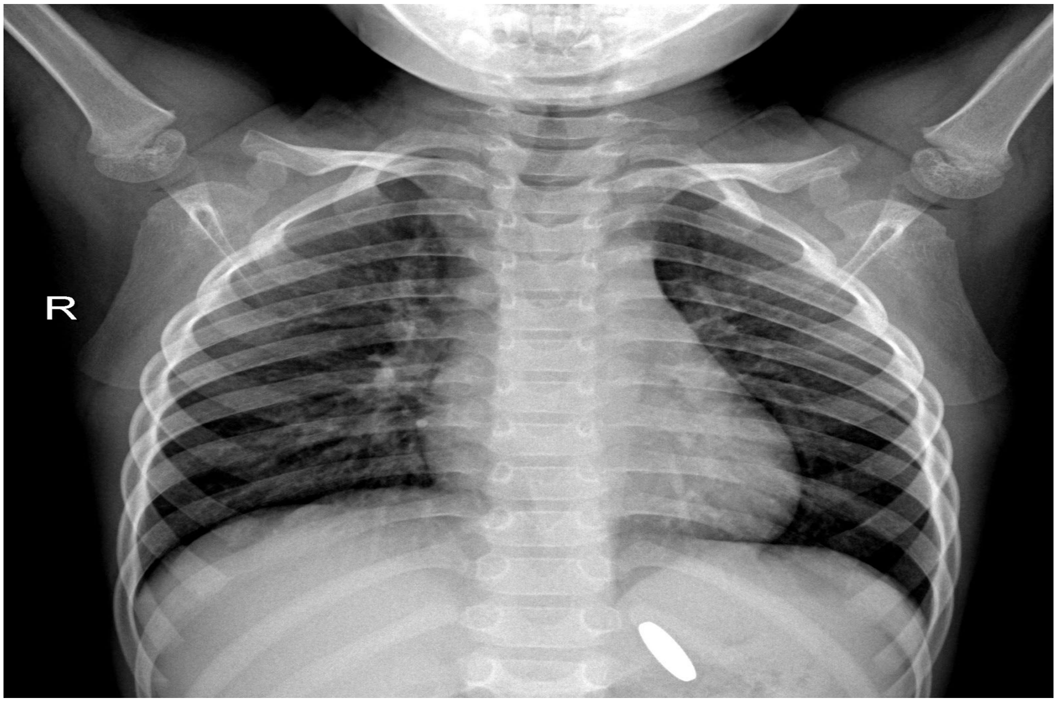


Image Classification of COVID-19 X-rays

CS 440 Class Challenge

12.04.2020

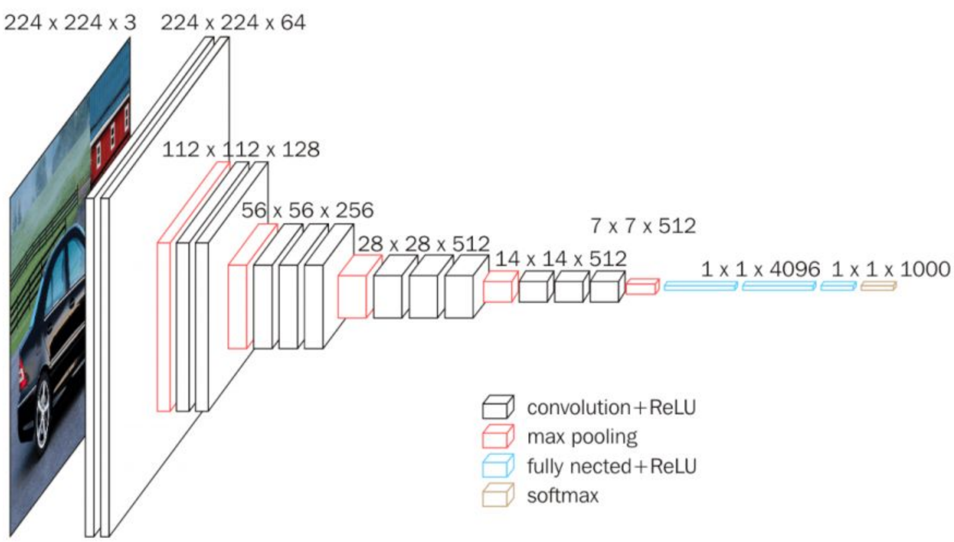
Giancarlo Sirio
U71347585
gsirio@bu.edu

BOSTON UNIVERSITY

Task 1

Architecture

Since the dataset was relatively small, an architecture transfer learning would be beneficial for the binary classification of X-rays. Based on trial and error, I decided to use the VGG-16 architecture proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition".



Visual Representation of VGG-16 Layers

For my model I used the VGG-16 implementation on Keras without its top layers and replaced the top layers starting from the Max Pooling layer. The layers following the Flatten Layer were as follows: A Flatten layer to reduce the dimensions of the weights, Fully Connected (Dense) layer with a ReLu activation function, Dropout Layer with a 0.25 dropout rate, and a final Fully Connected Layer with a singular output node and a Sigmoid activation function . All of the model's initial layers were frozen. The model was trained on the ImageNet dataset and its weights were downloaded before using it on this project.

Flatten -> Full Connected with ReLu -> Dropout (Rate= .25) -> Fully Connected with Sigmoid

`Additional Layers Used on Top of VGG-16 Model`

Loss & Regularization

For the **Loss Function** I used a **Binary Cross-Entropy** function due to my last Fully Connected Layer having a singular output node. For regularization the dataset was augmented by manipulating the dataset images in order to increase the number of test images that are available and reduce overfitting of the model.

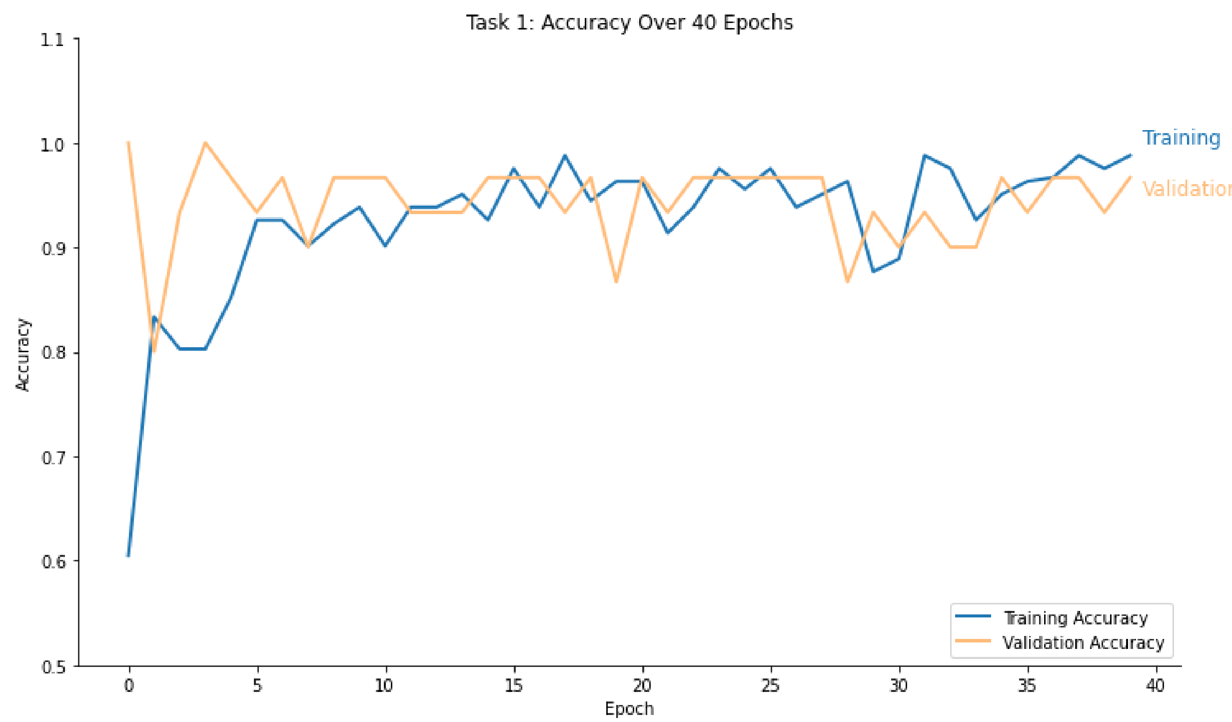
In addition the Dropout layer was used to reduce overfitting and help improve the generalizability of the model. The rate of nodes dropped was set to .25 which helped maintain a balance between the training and validation accuracy.

Other hyperparameters include

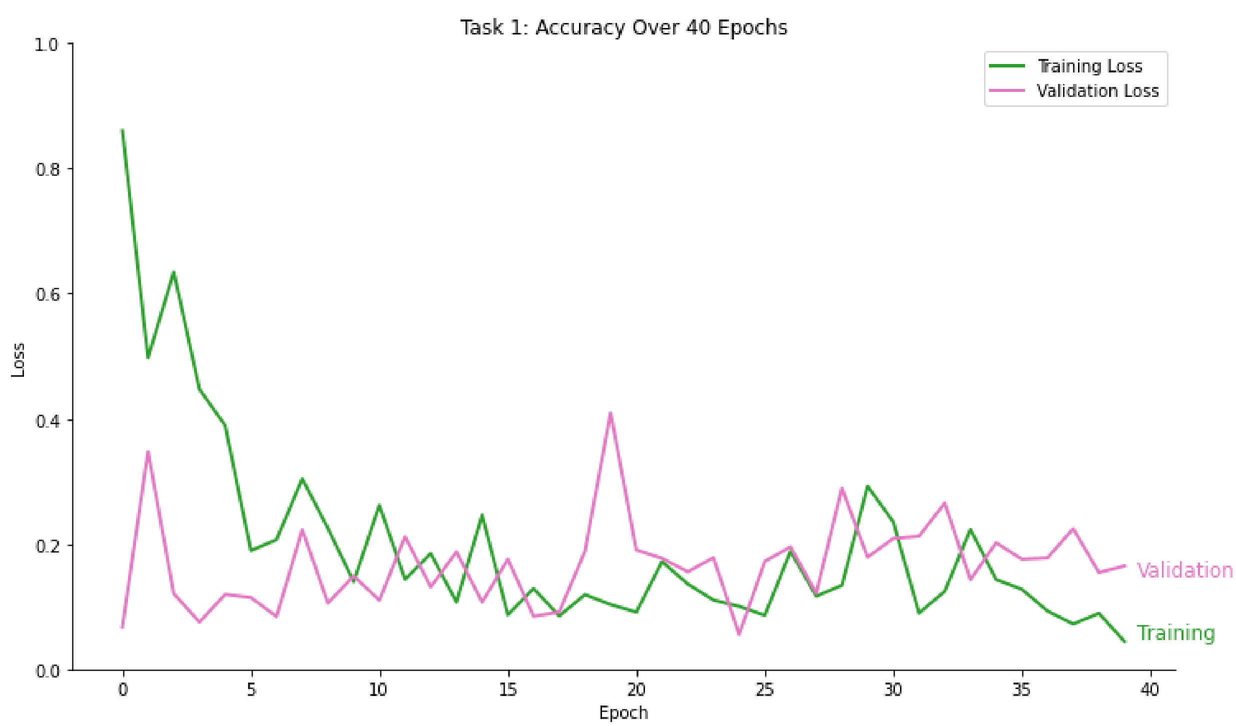
- Epochs : 40 Was gradually increased as the model was being tuned
- Learning Rate : .0001 Was gradually decreased as the model was being tuned
- Batch Size: 10

Training and Accuracy

A 70/30 Split of the data was used to train and validate the model.



From the graph we can see that the Training and Validation accuracies fluctuate heavily after about 10 epochs, but there is a steady positive trend in both metrics.



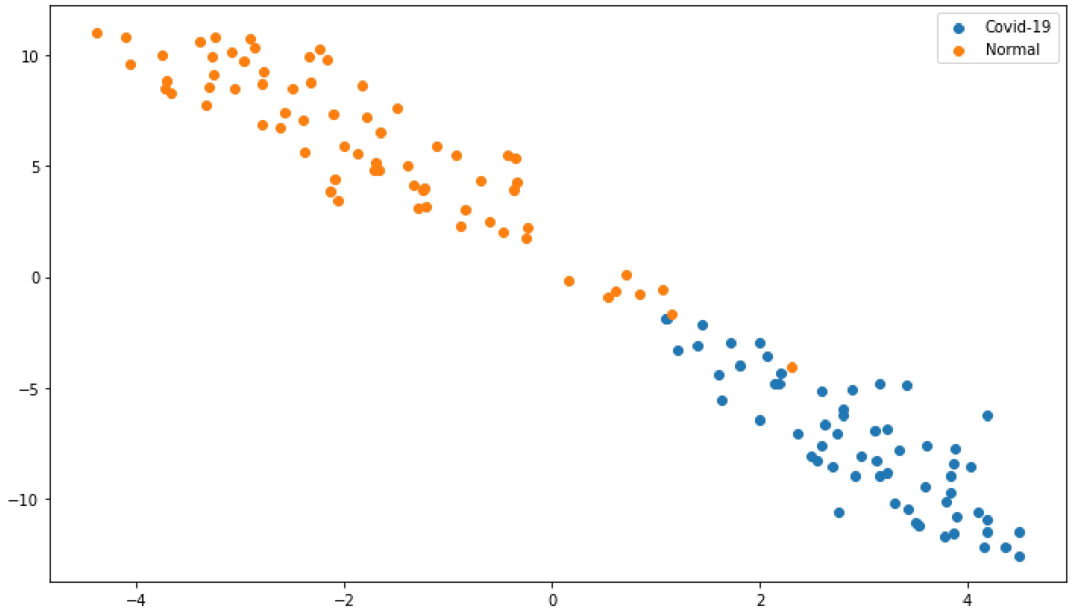
BOSTON UNIVERSITY

Here for the loss we see a steady negative trend in Training and Validation accuracy. Both metrics fluctuate at around 20 epochs but seem to become more steady after that.

After 40 Epochs of training the final metric were as follows :

Loss: 0.0401 - Accuracy: 0.9877 - Val_Loss: 0.1648 - Val_Accuracy: 0.9667

T-SNE Plot



Here we see that the model did a pretty good job at classifying the COVID-19 x-rays and the Normal x-rays. The clusters are pretty distinct but we do see a few points that are supposed to be Normal but were classified with an accuracy closer to being a Covid-19 positive case. In this setting a False-Positive diagnosis of Covid-19 would be preferred over a False-Negative diagnosis of being Covid-19.

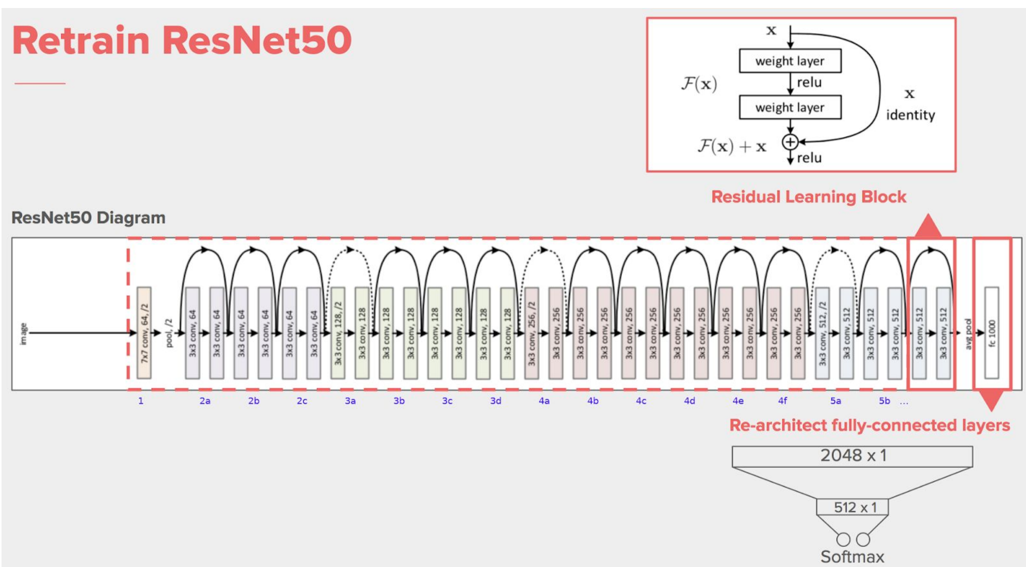
Task 2

Architectures:

Both Models were downloaded with pretrained weights from the ImageNet dataset.

Model 1: ResNet50

ResNet50 is a deep residual network that won the ILSVRC 2015 classification task. It is a 50 layers deep neural network. I modified the top layers in order to attempt to fine tune the parameters towards the multi-classification of the x-ray images. I add an AveragePoolin2D Layer with a pooling size of (7,7) and then I added a Flatten Layer. I added Dropout layers all with a rate of 0.5 followed before every Fully Connected (Dense) Layers all with ReLu activation function with the exception of the last layer that has a softmax activation function



Average Pooling 2D-> Flatten -> Dropout -> Dropout(.5) -> Fully Connected -> Dropout -> Fully Connected -> Fully Connected

The Added Top Layers to the Model

Model 2: Inception-V3

Inception-V3 is the third version of Google's Inception Convolutional Neural Network, which was first introduced during the ImageNet Recognition Challenge. I used this network without its top layers to fine tune the parameters on the x-rays data set. The added layers include a Global Average Pooling Layer, Fully Connected Layer, and one last Fully Connected Layer.

Average Pooling -> Fully Connected -> Fully Connected

Loss & Regularization Part II

For both Model 1 and Model 2 the data augmentation can be helpful to prevent overfitting. In both Models I used a Categorical Cross-Entropy Loss Function due to it being a multi-classification problem. For Model 1 I included various Dropout layers with rates of 0.05.

Other hyperparameters include

Model 1

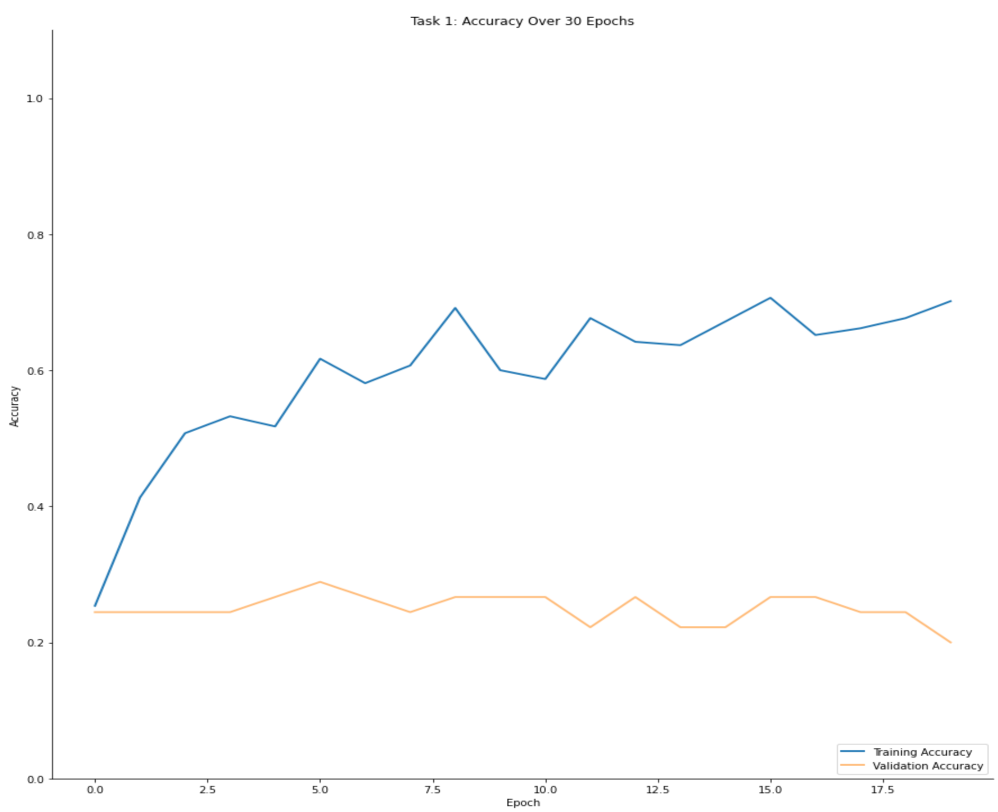
- Learning Rate = .0001
- Optimizer = Adam
- Epochs = 20

Model 2

- Epochs : 30 Was gradually increased as the model was being tuned
- Learning Rate : .001 Was gradually decreased as the model was being tuned
- Batch Size: 15

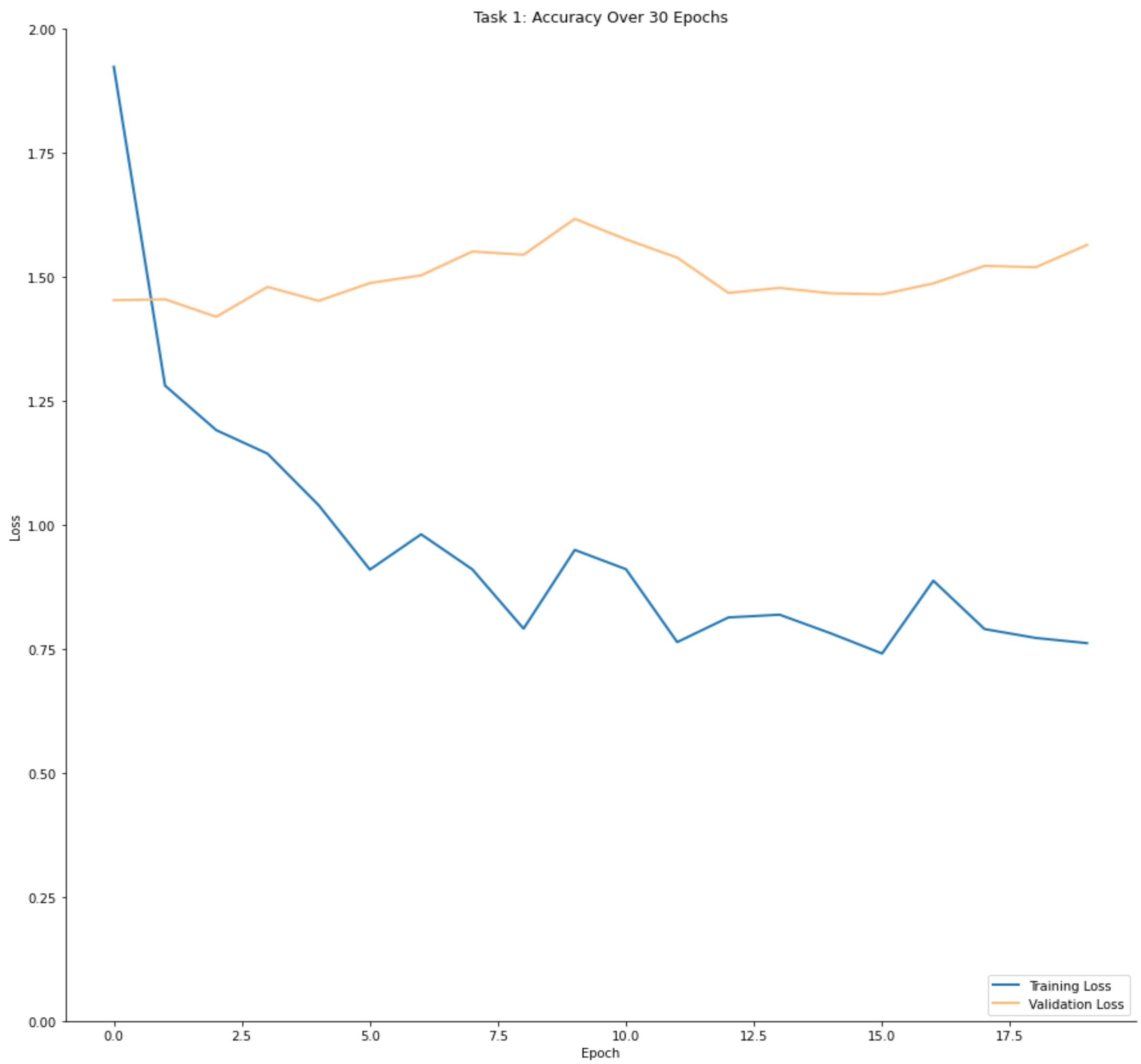
Training and Accuracy Part II

A 80/20 split was used for training of both models.



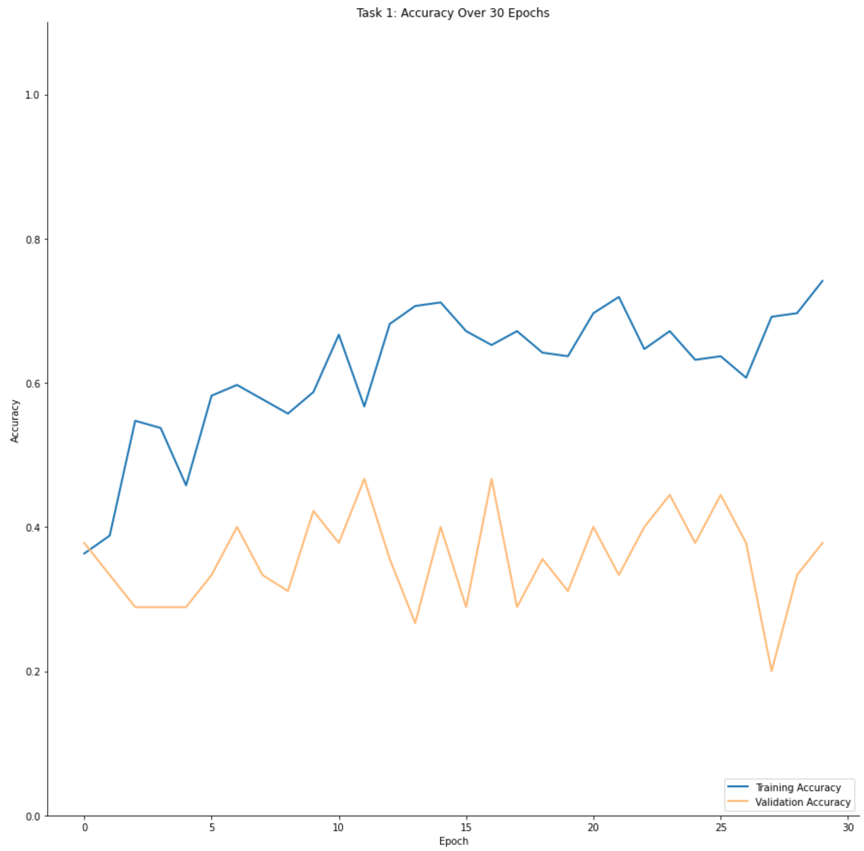
Model 1:

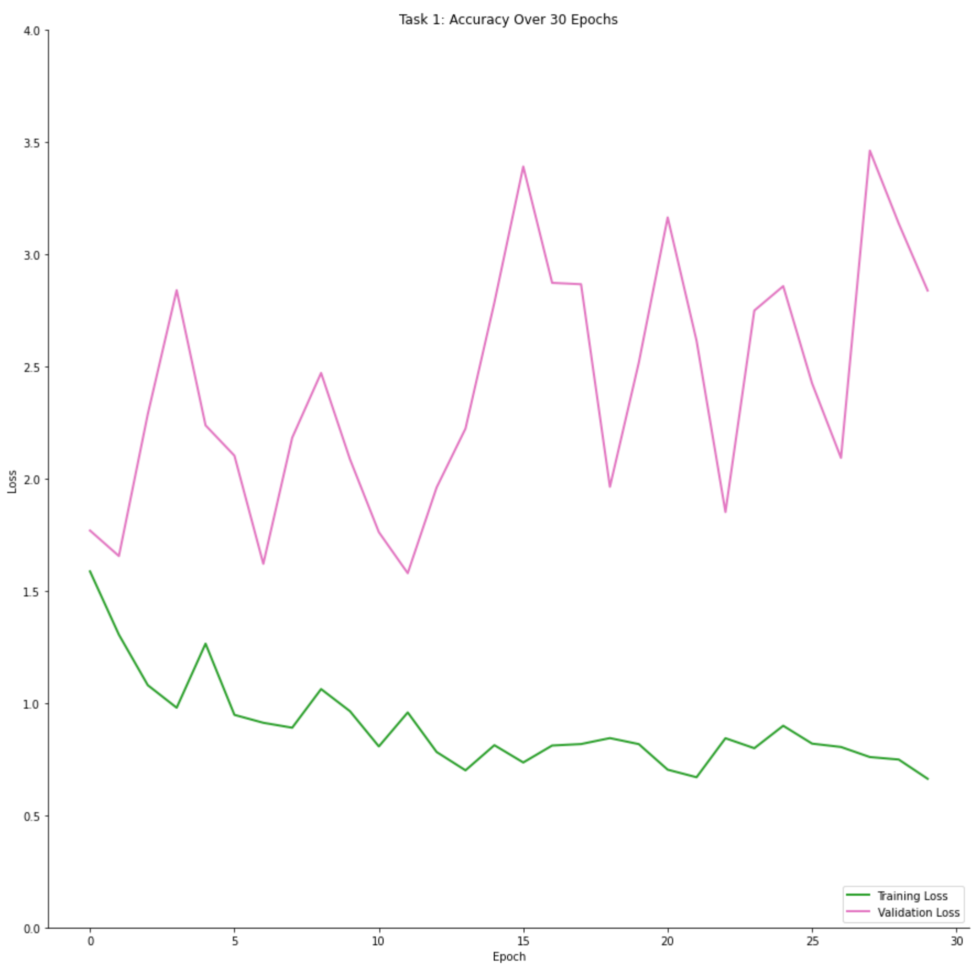
The accuracy graph on the right we can see the training accuracy increased every epoch while the validation accuracy stays relatively the same. This is evidence of overfitting because the model trains well on the training images but does not generalize well on the validation images.



After 20 epochs the final training/validation accuracy and loss was

loss: 0.7723 - accuracy: 0.7015 - val_loss: 1.5648 - val_accuracy: 0.2000





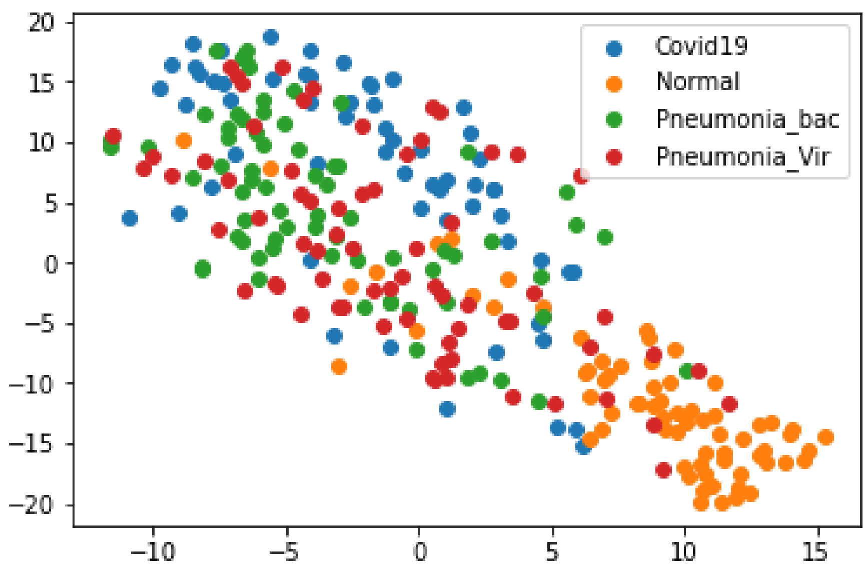
BOSTON UNIVERSITY

Here we can see that the training accuracy fluctuates over 30 epochs but the trend is overall positive. The validation accuracy is still not as close to the training accuracy so we can say that the model is overfitting to the training data. Heavier regularization might be considered for future training. The final metrics after 30 epochs was

loss: 0.6859 - accuracy: 0.7413 - val_loss: 2.8392 - val_accuracy: 0.3778

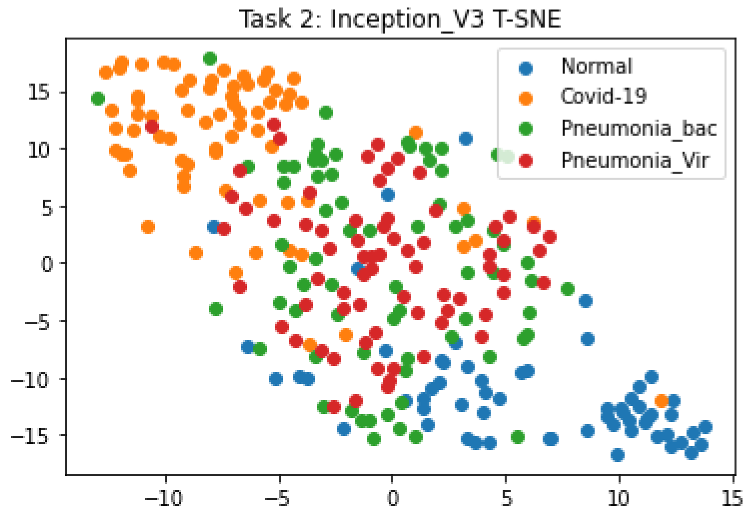
T-SNE Plot Part II

Model 1:



From this image we can see that the model does a good job of clustering together the Normal cases but is not able to separate the other 3 categories.

Model 2:



The Inception V3 model does a good job of clustering together the Normal and Covid19 cases but it does not do a good job of clustering the Pneumonia categories.

Conclusion and Improvements

Inception V3 the second model of choice performed overall better than the ResNet50 model. Inception V3 is a larger model overall which could have led to better trained feature detection from the ImageNet data set and have helped the transfer learning process with our X-ray dataset. One improvement I could see for these models is to train first on a larger set of x-rays first and then train again on our Covid-19 data set. This way the model could get time to familiarize itself with features common to all x-ray images and help with the lack of data overall. Another improvement could be to keep the model running for longer and potentially add more layers to the top of the network.