Maya Hussein

CSCE-4604

November 2022

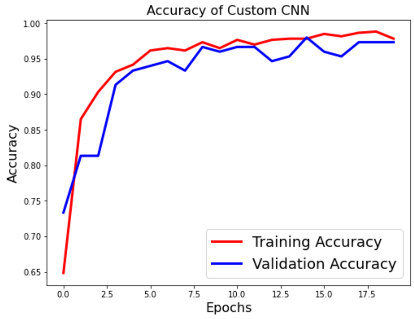
Lab Assignment 3

1 Introduction

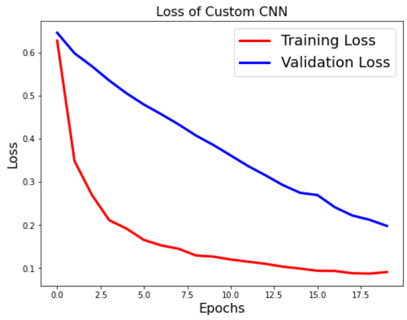
This lab assignment uses uses various types of CNN models to identify the emotional state of an individual (Happy or Not). A custom CNN model was designed along with the usage of multiple ready pretrained or not VGG, ResNet, and MobileNet. Below are the detailed description of each model.

2 Part I: Custom CNN Model

Below are the results of the Custom CNN Model. The model uses 16 filters with 5x5 kernel size, an relu activation function for non-linearity, maxpooling with size 2x2 and stride length of 1, and a sigmoid as the output funtion for binary classification. The results obtained from this model were an accuracy of 97.7% for both the training and validation and a loss of 19% for the validation and 9% the training data. When experimenting with different filter and kernel size the accuracy ranged from 89% to 93%. Below is a model with 89% accuracy with 3 filers with kernel size of 2x2. It is clear in figure (2) that the model greatly overfits in comparison to figure (1).

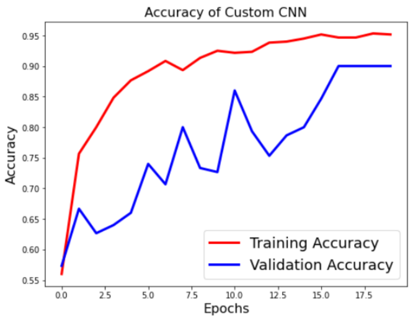


(a) Accuracy vs Epochs

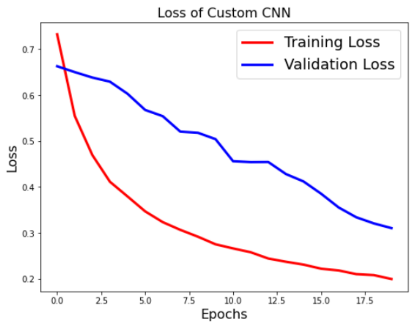


(b) Loss vs Epochs

Figure 1: CNN Model with 16 filters and 5x5 Kernel Size



(a) Accuracy vs Epochs



(b) Loss vs Epochs

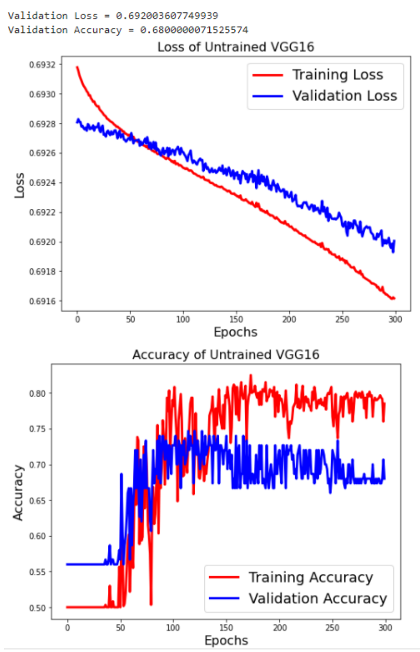
Figure 2: CNN Model with 3 filters and 2x2 Kernel Size

3 Part II: CNN Use Case No.1: VGG

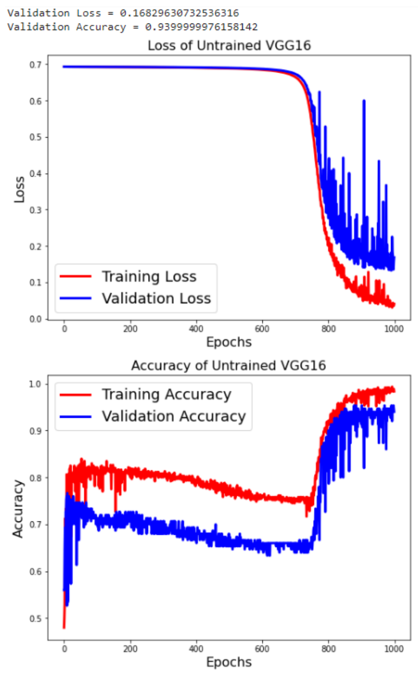
3.1 CNN Model: VGG16

3.1.1 Untrained Model

The below loss and accuracy vs. Epoch graphs are modeled using different number of Epochs. It is clear that the untrained VGG 16 Model in 300 epochs was overfitting as the gap between the training and validation is highly abridged. The model yielded an accuracy of 68% and a loss of 69.2%; the model does not converge to 0. However, when the number of epochs increased to a 1000, the accuracy drastically increased to 93.9% with a loss of only 16.8%. Even though model (b) does have irregularities in its graph, it is much better at 1000 epochs than 300. The pre-trained with and without freezing layers have yielded enhanced results.



(a) Untrained VGG-16 in 300 Epochs

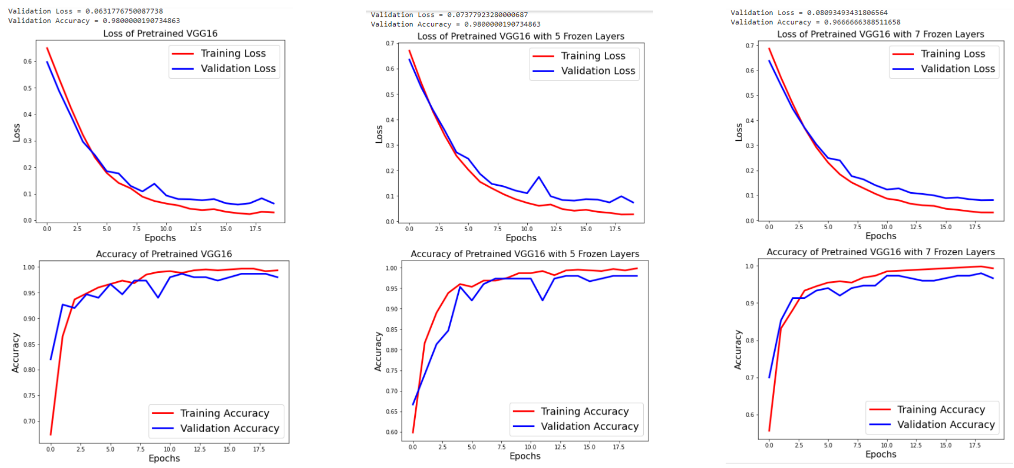


(b) Untrained VGG-16 in 1000 Epochs

Figure 3: Untrained VGG-16 Model

3.1.2 Pre-trained Model

The pre-trained VGG-16 was experimented on with 0, 5, and 7 freezing layers. The results of its loss and accuracy were 6.03% and 98%, 7.37% and 98%, and 8.09% and 96.6%, respectively. Even though the models (a) and (b) are similar in accuracies, model (a) has a lower cost(loss) making it a better solution than (b). Freezing seven layers has resulted in lower accuracy and greater loss than the other models making it the least feasible. As noted the accuracies were really high, so there was no need to increase the number of epochs for better accuracies. None of the models overfit as the validation and training accuracies are really close.



(a) Pre-Trained VGG-16 Model with No Freezing Layers

(b) Pre-Trained VGG-16 Model with 5 Freezing Layers

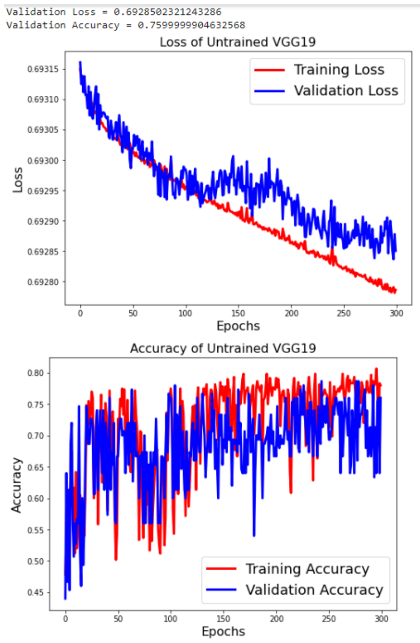
(c) Pre-Trained VGG-16 Model with 7 Freezing Layers

Figure 4: Pre-Trained VGG-16 Models

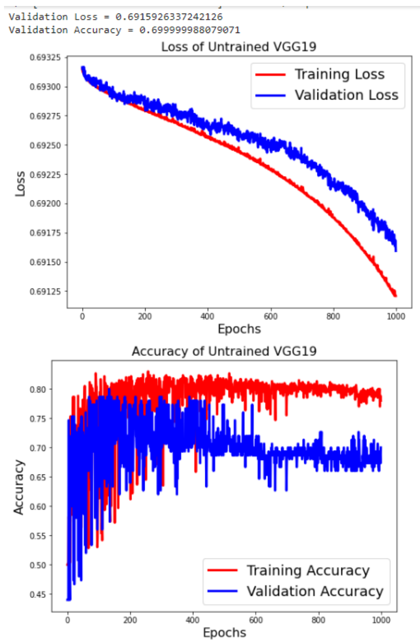
3.2 CNN Model: VGG19

3.2.1 Untrained Model

The below loss and accuracy vs. Epoch graphs are modeled using different number of Epochs. It is clear that the untrained VGG-19 Model in both epochs was dramatically overfitting. The models yielded an accuracy of 75.9% and a loss of 69.28% for model (a) and an accuracy of 69.99% and a loss of 69.15% for model (b); both models' losses do not converge to 0.



(a) Model with 300 Epochs

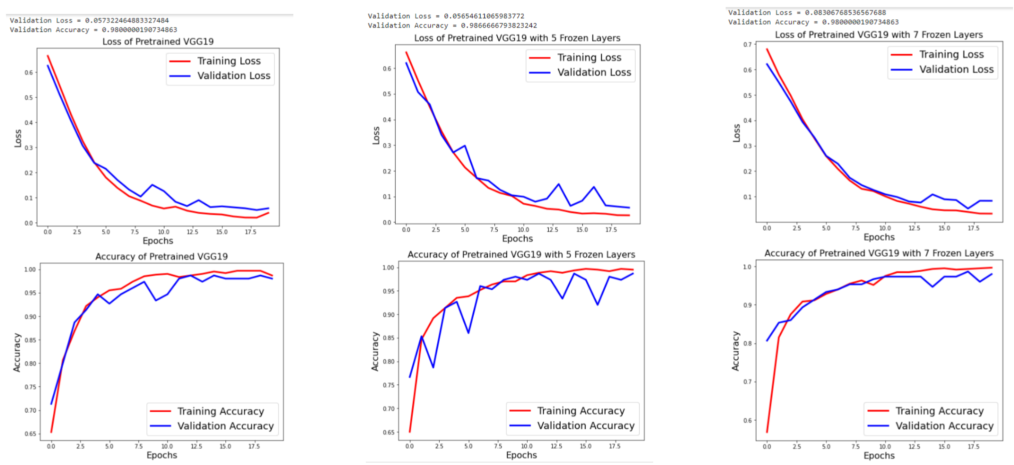


(b) Model with 1000 Epochs

Figure 5: Untrained VGG-19 Model

3.2.2 Pre-trained Model

The pre-trained VGG-19 was experimented on with zero, five, and seven freezing layers. The results of its loss and accuracy were 5.73% and 98%, 5.65% and 98.67%, and 8.306% and 98%, respectively. Even though the all the models are similar in accuracies, model (b) with 5 freezing layers has a lower cost(loss) and a greater accuracy making it a better solution than (a) and (c). As noted the accuracies were really high, so there was no need to increase the number of epochs for better accuracies. None of the models overfit as the validation and training accuracies are really close. However, it is good to note that model(b) has the most spikes in its graphs in comparison to models (a) and (c).



(a) Pre-Trained VGG-19 Model with No Freezing Layers

(b) Pre-Trained VGG-19 Model with 5 Freezing Layers

(c) Pre-Trained VGG-19 Model with 7 Freezing Layers

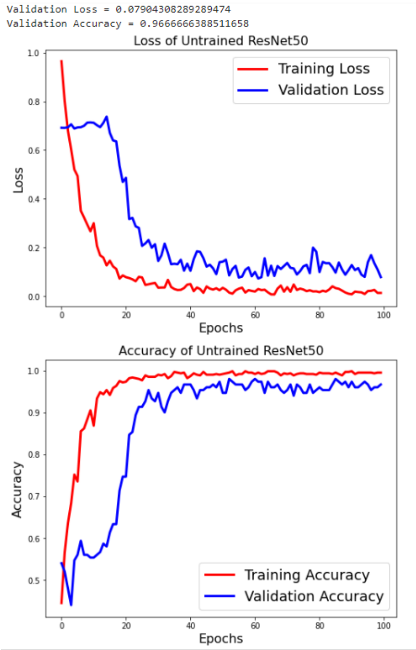
Figure 6: Pre-Trained VGG-19 Models

4 Part II: CNN Use Case No.2: Residual Connections-based Networks

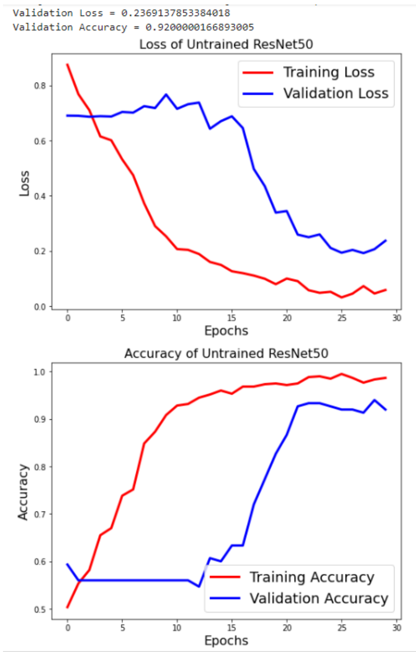
4.1 CNN Model: ResNet50

4.1.1 Untrained Model

The below loss and accuracy vs. Epoch graphs are modeled using different number of Epochs. It is clear that the untrained ResNet50 Model in 30 epochs was overfitting as the gap between the training and validation is highly abridged. The model yielded an accuracy of 92% and a loss of 23.69%; the model does not converge to 0. However, when the number of epochs increased to a 100, the accuracy increased to 96.6% with a loss of only 7.904%. Even though model (a) does have irregularities in its graph, it is much better at 100 epochs than 30. The pre-trained with and without freezing layers have yielded enhanced results.



(a) Model with 100 Epochs

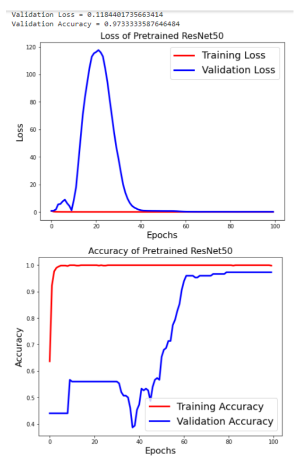


(b) Model with 30 Epochs

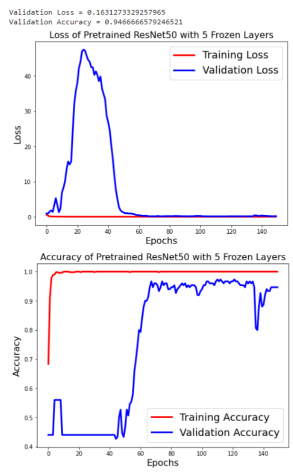
Figure 7: Untrained ResNet50 Model

4.1.2 Pre-trained Model

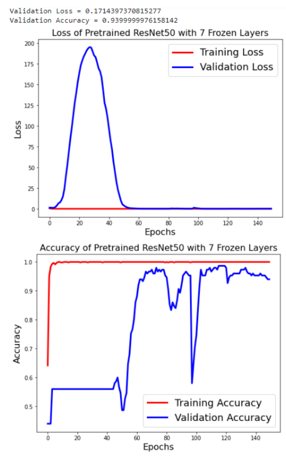
The pre-trained ResNet50 was experimented on with zero, five, and seven freezing layers. The results of its loss and accuracy were 11.84% and 97.33%, 16.31% and 94.66%, and 17.14% and 93.99%, respectively. All models have relatively close accuracies; however, model (a) has the best accuracy. All models have great overfitting in the beginning of the epoch. Other models generated better graphs and accuracies.



(a) Pre-Trained ResNet50 Model with No Freezing Layers



(b) Pre-Trained ResNet50 Model with 5 Freezing Layers



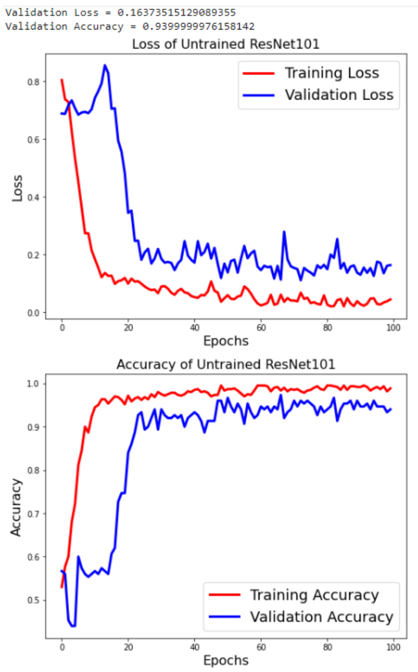
(c) Pre-Trained ResNet50 Model with 7 Freezing Layers

Figure 8: Pre-Trained ResNet50 Models

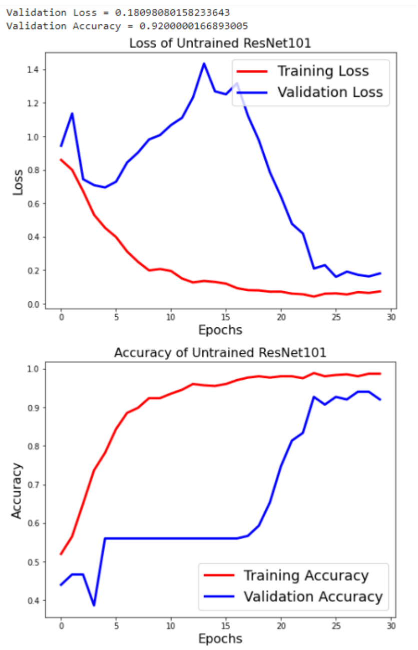
4.2 CNN Model: ResNet101

4.2.1 Untrained Model

The below loss and accuracy vs. epoch graphs are modeled using different number of Epochs. It is clear that the untrained ResNet50 Model in 30 epochs was overfitting as the gap between the training and validation is highly abridged. The model yielded an accuracy of 92% and a loss of 18.09%; the model does not converge to 0. However, when the number of epochs increased to a 100, the accuracy increased to 93.99% with a loss of only 16.37%. Even though model (a) does have irregularities in its graph, it is much better at 100 epochs than 30. It is clear that model(b) overfits a lot more than model(a). The pre-trained with and without freezing layers have yielded enhanced results.



(a) Model with 100 Epochs

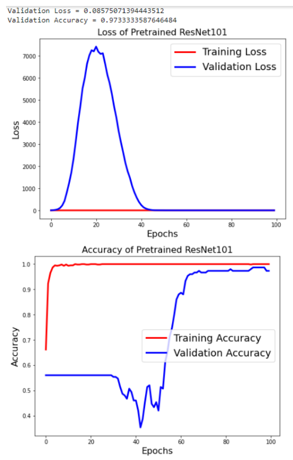


(b) Model with 30 Epochs

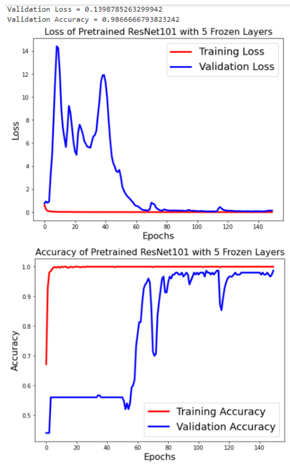
Figure 9: Untrained ResNet101 Models

4.2.2 Pre-trained Model

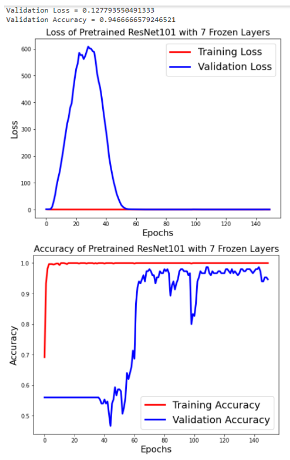
The pre-trained ResNet101 was experimented on with zero, five, and seven freezing layers. The results of its loss and accuracy were 8.57% and 97.33%, 13.98% and 98.66%, and 12.77% and 94.66%, respectively. All models have relatively close accuracies; however, model (b) has the best accuracy. All models have great overfitting in the beginning of the epoch. Other models generated better graphs and accuracies.



(a) Pre-Trained ResNet101 Model with No Freezing Layers



(b) Pre-Trained ResNet101 Model with 5 Freezing Layers



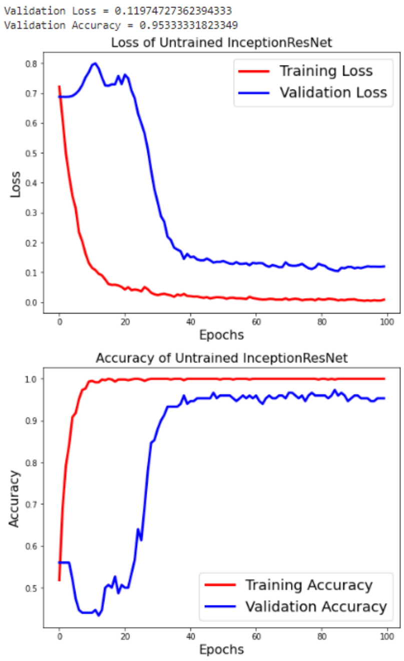
(c) Pre-Trained ResNet101 Model with 7 Freezing Layers

Figure 10: Pre-Trained ResNet101 Models

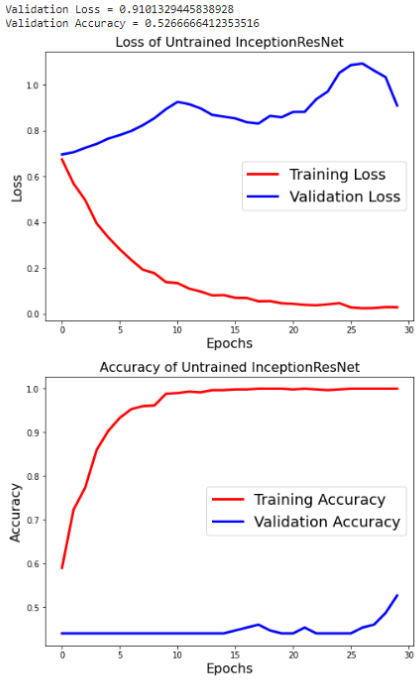
4.3 CNN Model: InceptionResNet

4.3.1 Untrained Model

The below loss and accuracy vs. epoch graphs are modeled using different number of Epochs. It is clear that the untrained Inception ResNet Model in 30 epochs was overfitting as the gap between the training and validation is highly abridged. The model yielded an accuracy of 52% and a loss of 91.01%; the model does not converge to 0. However, when the number of epochs increased to a 100, the accuracy increased to 95.33% with a loss of only 11.97%. Even though model (a) does have irregularities in its graph, it is much better at 100 epochs than 30. It is clear that model(b) overfits a lot more than model(a). The pre-trained with and without freezing layers have yielded enhanced results.



(a) Model with 100 Epochs

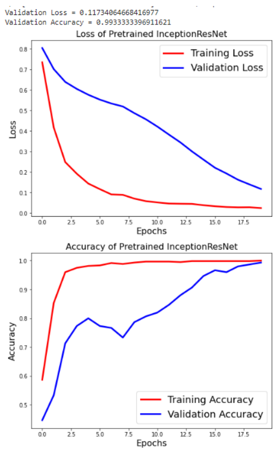


(b) Model with 30 Epochs

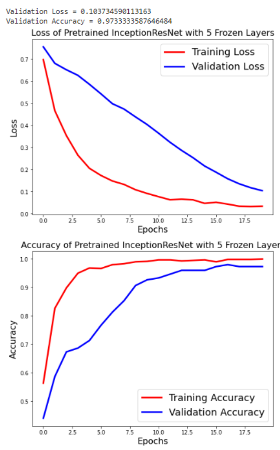
Figure 11: Untrained Inception ResNet Models

4.3.2 Pre-trained Model

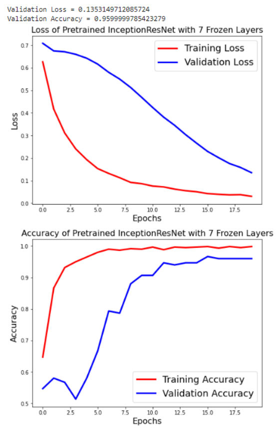
The pre-trained InceptionResNet was experimented on with zero, five, and seven freezing layers. The results of its loss and accuracy were 11.73% and 99.33%, 10.37% and 97.33%, and 13.53% and 95.99%, respectively. All models have relatively close accuracies; however, model (a) has the best accuracy. All models have great overfitting in the beginning of the epoch.



(a) Pre-Trained Inception ResNet Model with No Freezing Layers



(b) Pre-Trained Inception ResNet Model with 5 Freezing Layers



(c) Pre-Trained Inception ResNet Model with 7 Freezing Layers

Figure 12: Pre-Trained Inception ResNet Models

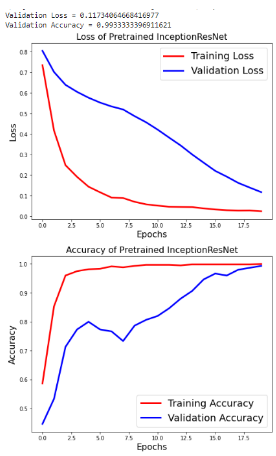
4.4 Inception ResNet

As noticed the Inception ResNet has yielded the best accuracy and even though its loss is around 11%. Hence, it will be the model used for optimization.

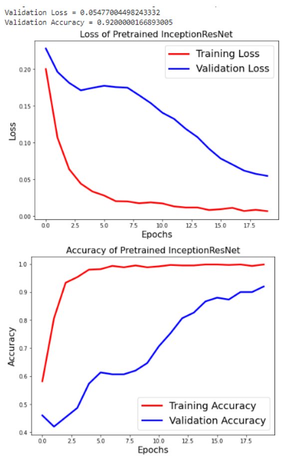
4.4.1 Loss Functions

Binary Cross Entropy and Binary Focal Cross entropy were utilized. The results were 99.33% for the binary cross entropy and 92% for Binary Focal Cross entropy. Even both models overfit, model (b) is much worse. No other loss function can be used as this is a classification problem so regression losses cannot be used, and all other probabilistic losses are used for multiple classes

(2 or more).



(a) Pre-Trained Inception ResNet Model with Binary Cross Entropy

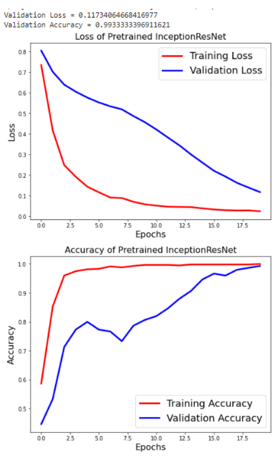


(b) Pre-Trained Inception ResNet Model with Binary Focal Cross Entropy

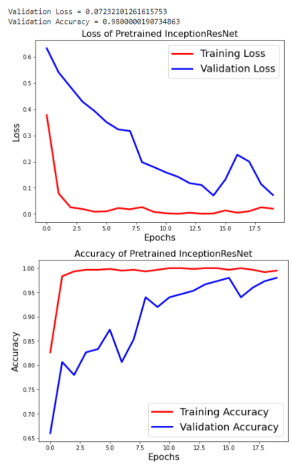
Figure 13: Pre-Trained Inception ResNet Models with Loss Functions

4.4.2 Optimizers

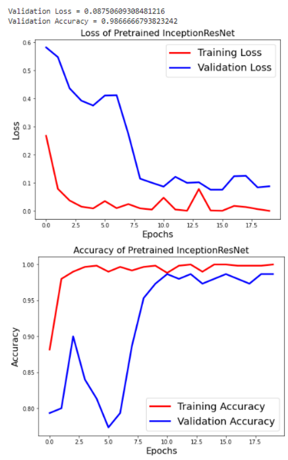
SGD, Adam, and RMSProp were used as optimizers when training the model. SGD yielded the highest accuracy of 99.3%, while Adam and RMSProp produced a result of 98% and 98.66%, respectively.Though all models overfit, model(c) with RMSProp has the greatest overfitting in the beginning of its epochs.



(a) Pre-Trained Inception ResNet Model with SGD



(b) Pre-Trained Inception ResNet Model with Adam



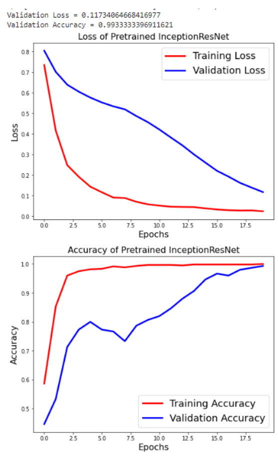
(c) Pre-Trained Inception ResNet Model RMSProp

Figure 14: Pre-Trained Inception ResNet Models with Optimizers

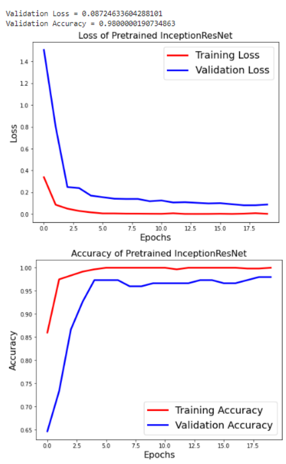
4.4.3 HyperParameters: Learning Rate Momentum

Three learning rates were tested: 0.0001, 0.003, and 0.000001, The results were 99.33%, 98%, and 92%, respectively. It is noticed that model(c) suffers from great overfitting when used with an 0.00001 learning rate.

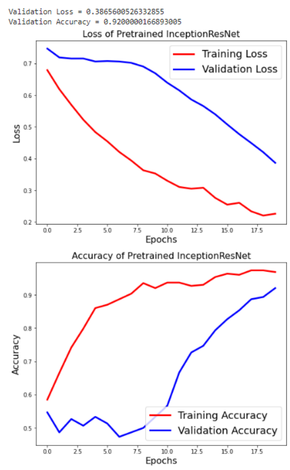
Three momentum rates were tested: 0.9, 0.3, and 1.5, A 1.5 momentum generated an error since the momentum has to be between 0 and 1. The other models' results were 99.33% for the 0.9 and 90% for the 0,3 momentum. It is noticed that model(b) suffers from great overfitting when used with an 0.3 momentum.



(a) Pre-Trained Inception ResNet Model with 0.0001 LR



(b) Pre-Trained Inception ResNet Model with 0.003 LR

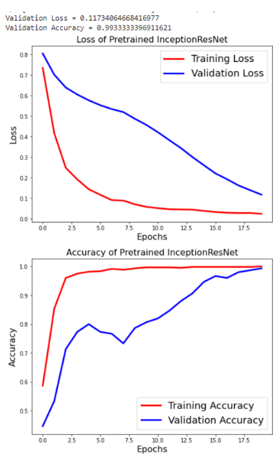


(c) Pre-Trained Inception ResNet Model with 0.00001 LR

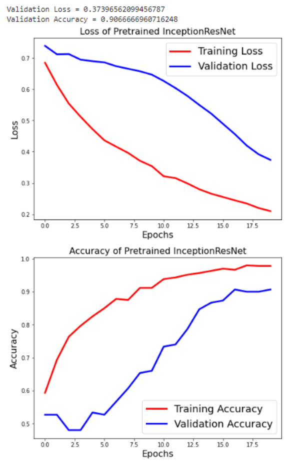
Figure 15: Pre-Trained Inception ResNet Models with Different Learning Rates

5 CNN Use Case No.3: MobileNet

MobileNet is a general model that can be used for several cases. It uses different input layer sizes and width factors, which helps in reducing the inference cost on mobile devices. It generally supports input sizes with size greater than 32 and offers better performance with larger image sizes. Because out image size is 64, MobileNet is a good solution to targeting our problem. https://www.tensorflow.org/api_docs/python/tf/keras/applications/mobilenet



(a) Pre-Trained Inception ResNet Model with 0.9 Momentum

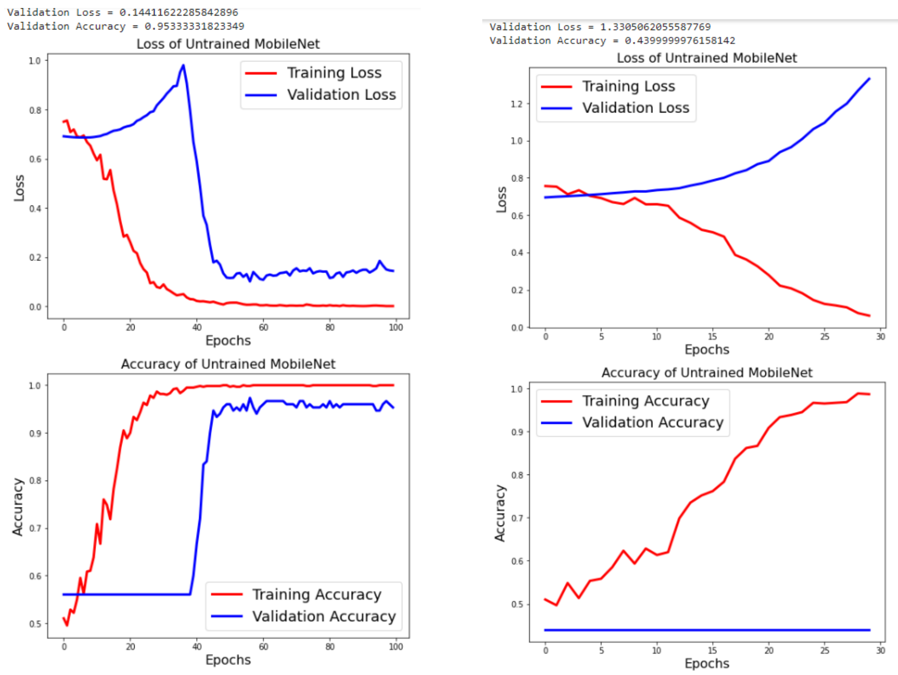


(b) Pre-Trained Inception ResNet Model with 0.3 Momentum

Figure 16: Pre-Trained Inception ResNet Models

5.1 Untrained Model

The below loss and accuracy vs. Epoch graphs are modeled using different number of Epochs. It is clear that the untrained MobileNet Model overfits greatly, and that the 30 epoch model is just a subsection of model (a) the 100 epochs. Model(b) yielded an accuracy of 43.99% and a loss of 133.05%; the model does not converge to 0. However, when the number of epochs increased to a 100, the accuracy drastically increased to 95.33% with a loss of only 14.41%. Even though model (a) does have irregularities in its graph, it is much better at 100 epochs than 30. It is clear that model(b) overfits a lot more than model(a). The pre-trained with and without freezing layers have yielded enhanced results.

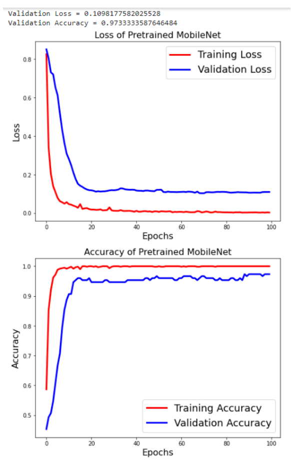


(a) Model with 100 Epochs (b) Model with 30 Epochs

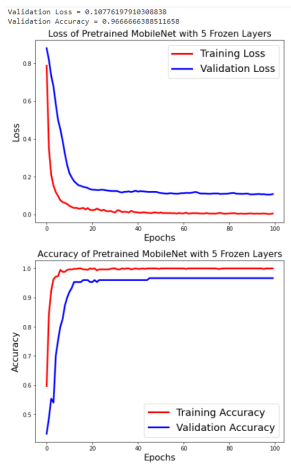
Figure 17: Untrained MobileNet

5.2 Pre-trained Model

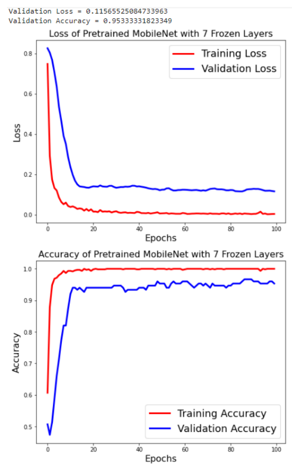
The pre-trained MobileNet was experimented on with zero, five, and seven freezing layers. The results of its loss and accuracy were 10.98% and 97.33%, 10.77% and 96.66%, and 11.56% and 95.33%, respectively. All models have relatively close accuracies; however, model (a) has the best accuracy. It is good to note that none of the graphs overfit as gap between the validation and training is not that wide.



(a) Pre-Trained MobileNet Model with No Freezing Layers



(b) Pre-Trained MobileNet Model with 5 Freezing Layers



(c) Pre-Trained MobileNet Model with 7 Freezing Layers

Figure 18: Pre-Trained VGG-19 Models