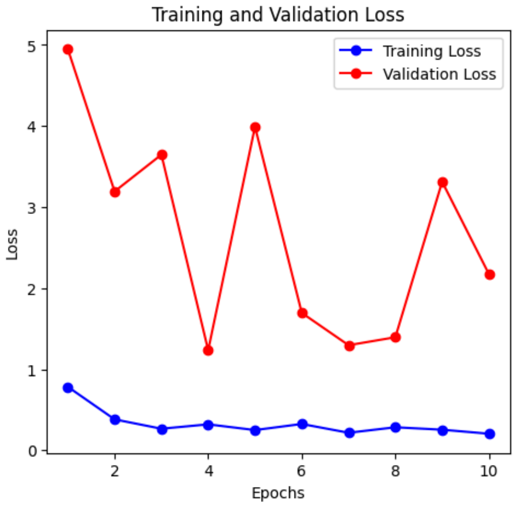
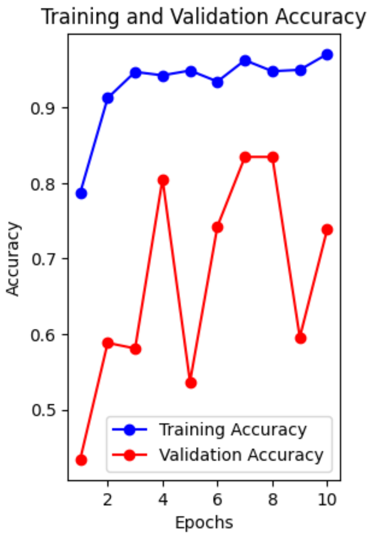
For this project, we built a Convolutional Neural Network (CNN) using the TensorFlow and Keras libraries to classify images of different monkey species from a Kaggle Dataset.

<table>
<tr><th>Model</th><th>Hyperparameters</th><th>Loss</th><th>Accuracy</th></tr>
<tr><td>1</td><td>No. of Layers = 4
Activation Function = ReLu
Epochs = 10</td><td>Training: 0.2043
Validation: 2.1731</td><td>Training: 0.9709
Validation: 0.7390</td></tr>
<tr><td colspan="4">

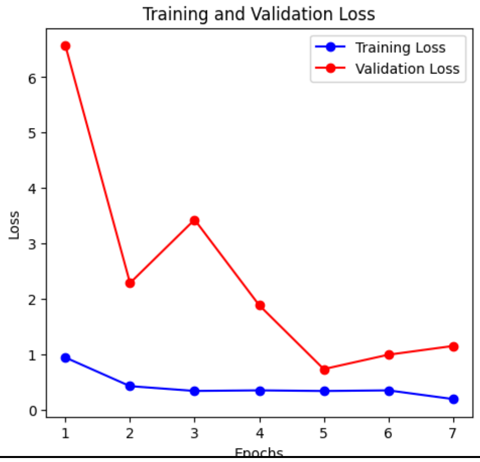


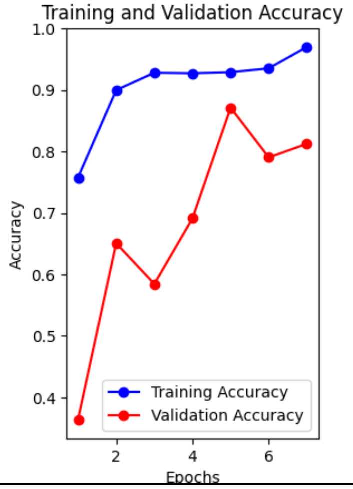


Based on the above plots, we can see that the model is optimal at epoch 7/8. For the next iteration we will reduce the epoch to prevent overfitting and limit unnecessary model complexity.

</td></tr>
<tr><td>2</td><td>No. of Layers = 5
Activation Function = ReLu
Epochs = 7</td><td>Training: 0.1925
Validation: 1.1500</td><td>Training: 0.9699
Validation: 0.8125</td></tr>
<tr><td colspan="4">

For this model, we added a dense layer and reduced the epochs to 7. The activation function

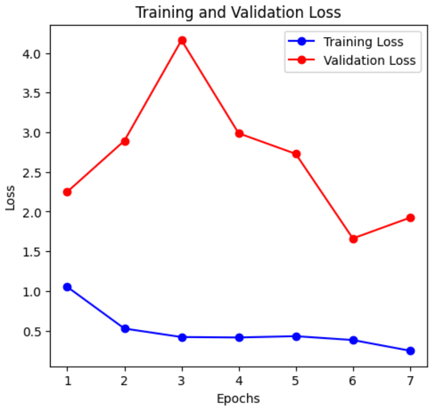


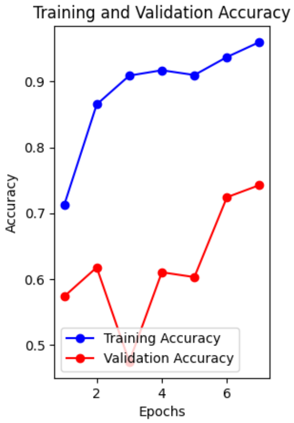


</td></tr>
</table>

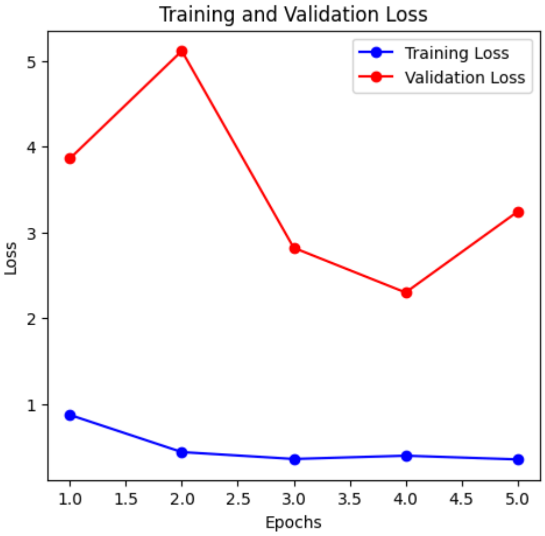
<table>
<tr><td colspan="4">remained the same.

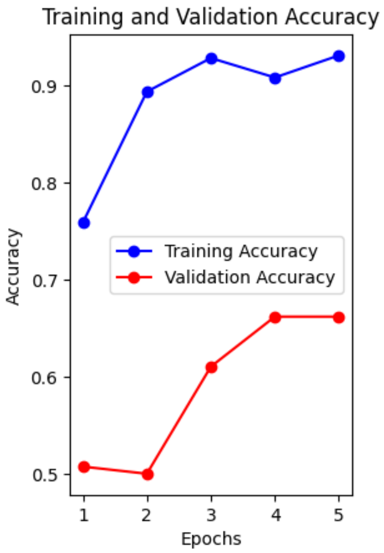
Based on the validation accuracy and loss for model 2, we can see that the second model performed better. Furthermore, the plots indicate that the model reaches its optimal performance around epoch 5, suggesting that further training beyond this point would not yield significant improvements and could potentially introduce overfitting.</td></tr>
<tr><td>3</td><td>No. of Layers = 7
Activation Function = ReLu
Epochs = 7</td><td>Training: 0.2478
Validation: 1.9237</td><td>Training: 0.9599
Validation: 0.7426</td></tr>
<tr><td colspan="4">For model 3, we added 2 more dense layers to see if this would increase the performance of the model as it did for model 2 when we added one dense layer.





After increasing the number of dense layers by 2 we can see the validation accuracy go down from the previous model. This indicates probable overfitting for model 3, so for the next iteration, we will return to 5 layers.</td></tr>
<tr><td>4</td><td>No. of Layers = 5
Activation Function = Leaky ReLu
Epochs = 5</td><td>Training: 0.3544
Validation: 3.2410</td><td>Training: 0.9308
Validation: 0.6618</td></tr>
<tr><td colspan="4">For this iteration, we used 5 layers as that worked the best in our previous models and changed the activation function to Leaky ReLu.</td></tr>
</table>





The validation accuracy for model 4 was lower than the accuracy for model 2, which was essentially the same hyperparameters except with leaky relu as the activation function.

Thus, from the models and hyperparameters tested, the best model was model 2.

We then chose 25 random images to test our model. Out of the 25 chosen, the model correctly predicted 16 of them. This gave us an accuracy of 64% for the selected images.