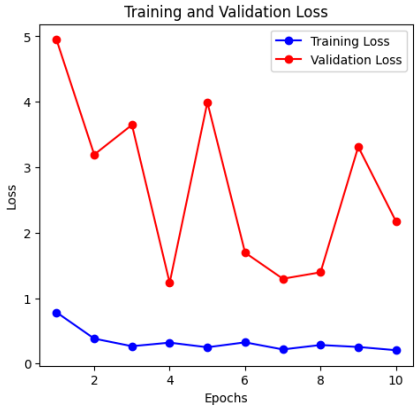
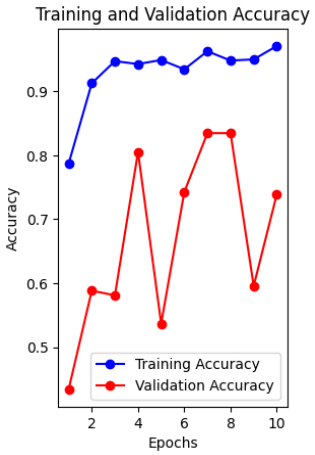
For this project, we built a Convolutional Neural Network (CNN) using the TensorFlow and Keras libraries to classify images of different monkey species from a Kaggle Dataset. There weren't any corrupted or files so no cleaning was needed. We performed one hot encoding on the labels.

Model	**Hyperparameters**	**Loss**	**Accuracy**
1	No. of Layers = 4 Activation Function = ReLu Epochs = 10	Training: 0.2043 Validation: 2.1731	Training: 0.9709 Validation: 0.7390

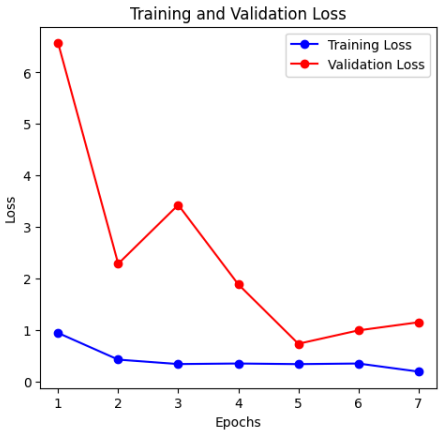


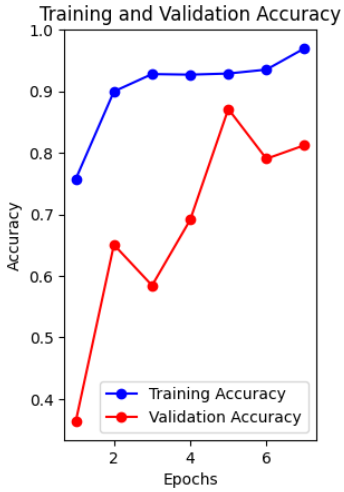


Based on the above plots, we can see that the model is optimal at epoch 7/8. For the next iteration we will reduce the epoch to prevent overfitting and limit unnecessary model complexity.

2	No. of Layers = 5 Activation Function = ReLu Epochs = 7	Training: 0.1925 Validation: 1.1500	Training: 0.9699 Validation: 0.8125

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



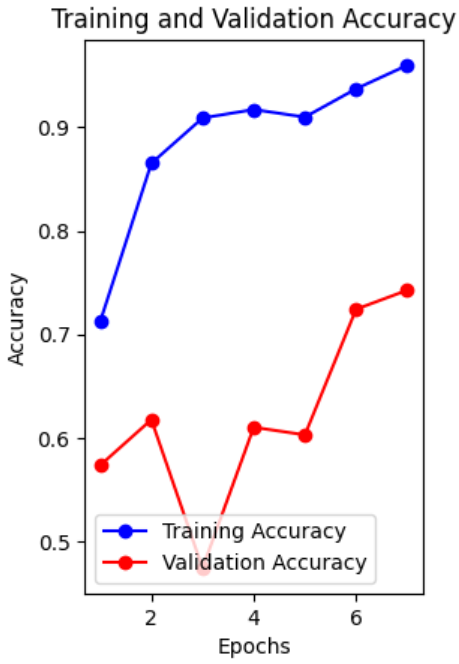


remained the same.

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.

3	No. of Layers = 7 Activation Function = ReLu Epochs = 7	Training: 0.2478 Validation: 1.9237	Training: 0.9599 Validation: 0.7426

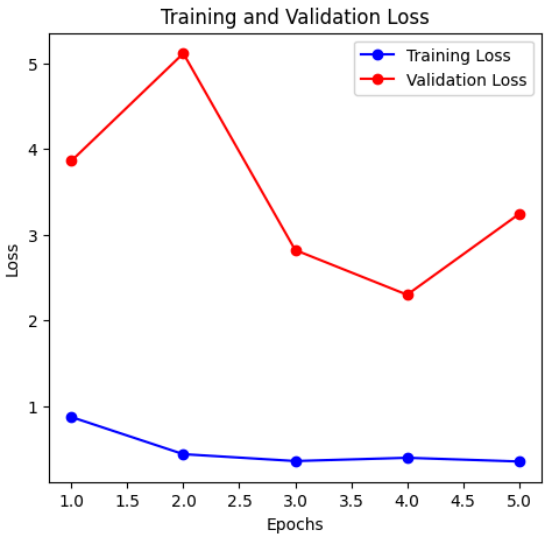
For model 3, we added 2 more dense layers to see if this would increase the performance of the model as it did for model 2 when we added one dense layer.

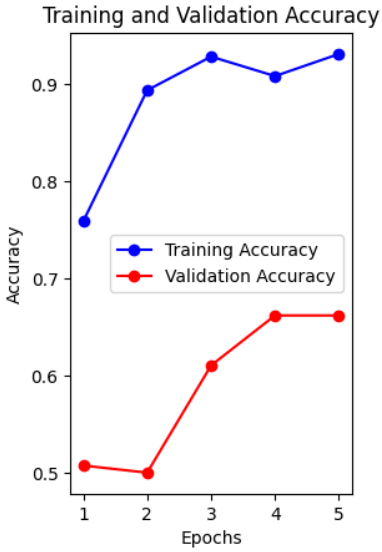


After increasing the number of dense layers by 2 we can see the validation accuracy go down from the previous model. This indicates probable overfitting for model 3, so for the next iteration, we will return to 5 layers.

4	No. of Layers = 5 Activation Function = Leaky ReLu Epochs = 5	Training: 0.3544 Validation: 3.2410	Training: 0.9308 Validation: 0.6618

For this iteration, we used 5 layers as that worked the best in our previous models and changed the activation function to Leaky ReLu.

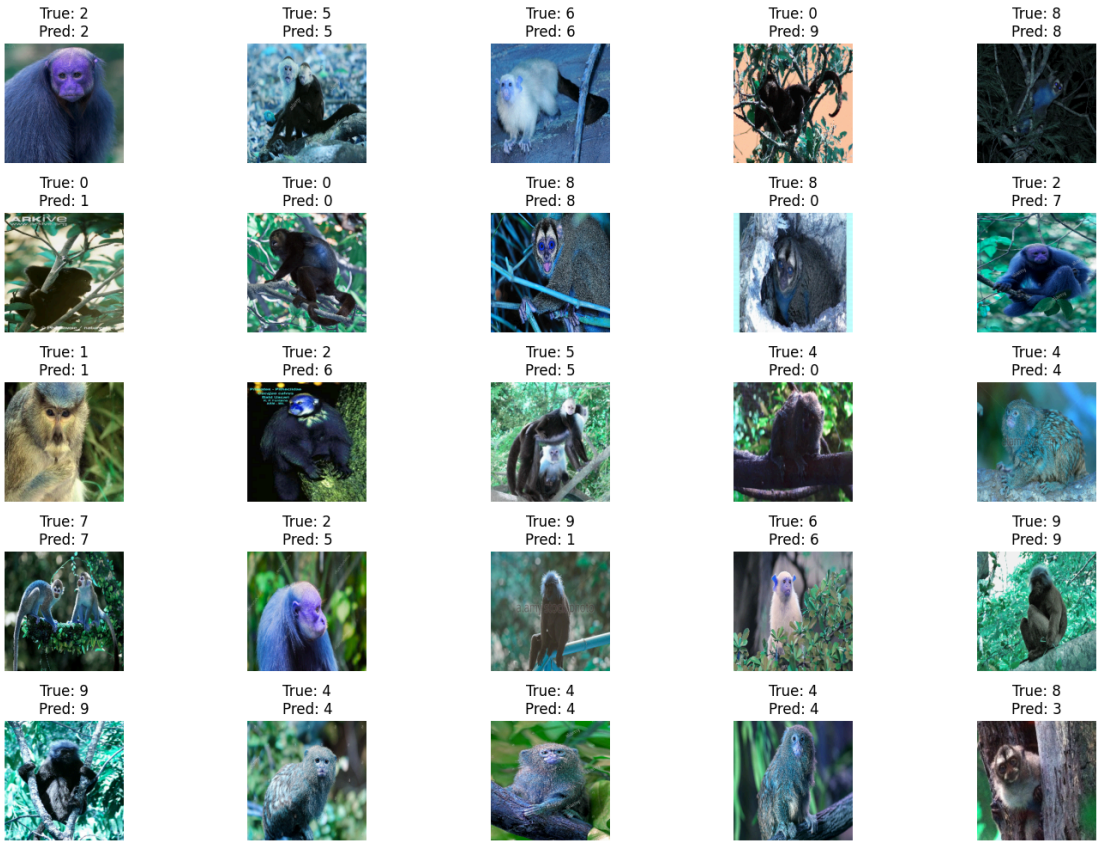




The validation accuracy for model 4 was lower than the accuracy for model 2, which was essentially the same hyperparameters except with leaky relu as the activation function.

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.



Conclusion:
In this project, we experimented with multiple variations of Convolutional Neural Networks to classify images of different monkey species. By adjusting the number of layers, activation functions, and epochs, we found that model 2 with 5 layers, ReLU activation, and 7 epochs provided the best balance between accuracy and overfitting. Despite working with a relatively small dataset, the model achieved an accuracy of 64% on a random sample of images, demonstrating its effectiveness given the constraints.

Future improvements could include data augmentation to expand our small dataset and trying more advanced models. Overall the project showed CNNs flexibility and power when balanced correctly.