Assignment 4

-Aditya Sankranthi

Task 1:

(i) Description of CNN Architectures:

- **Model 1:**
  - Architecture: Convolutional Neural Network with 2 convolutional layers followed by max pooling, then a flatten layer, a dense layer with ReLU activation, a dropout layer for regularization, and finally, an output layer with softmax activation.
  - Architecture Overview:
    - Convolutional Layer 1: 32 filters, kernel size (3,3), ReLU activation.
    - MaxPooling Layer: Pool size (2,2).
    - Convolutional Layer 2: 64 filters, kernel size (3,3), ReLU activation.
    - MaxPooling Layer: Pool size (2,2).
    - Flatten Layer: Flatten the output for dense layers.
    - Dense Layer: 128 neurons, ReLU activation.
    - Dropout Layer: Dropout rate of 0.5.
    - Output Layer: 10 neurons (corresponding to the 10 classes), softmax activation.

- **Model 2:**
  - Architecture: Convolutional Neural Network with 3 convolutional layers followed by max pooling, then a flatten layer, two dense layers with ReLU activation, a dropout layer for regularization, and finally, an output layer with softmax activation.
  - Architecture Overview:
    - Convolutional Layer 1: 16 filters, kernel size (3,3), ReLU activation.
    - MaxPooling Layer: Pool size (4,4).
    - Convolutional Layer 2: 32 filters, kernel size (3,3), ReLU activation.
    - MaxPooling Layer: Pool size (2,2).
    - Convolutional Layer 3: 256 filters, kernel size (3,3), ReLU activation.
    - Flatten Layer: Flatten the output for dense layers.
    - Dense Layer 1: 128 neurons, ReLU activation.
    - Dropout Layer: Dropout rate of 0.3.
    - Dense Layer 2: 10 neurons (corresponding to the 10 classes), softmax activation.

(ii) Training Accuracy Over Epochs:

Epoch	Model 1 Accuracy	Model 2 Accuracy
1	17.36%	24.76%
5	61.42%	67.10%
10	80.20%	84.49%
15	85.41%	90.79%
20	89.56%	92.56%
25		93.11%

(iii) Test Accuracy:

Test accuracy for Model 1	Test accuracy for Model 2
54.25%	71.02%

(iv) Confusion Matrix for Model 2:

```
[[ 52   0   0   0   0   0   0   0   0   0]
 [  0 118   4   7   3   7   0   2   0   5]
 [  1   2  20   4   1   0   2   3   2   0]
 [  0   3   2  42   7   1   0   1  16   1]
 [  1   0   1   0  25   0   0   1   3   3]
 [  0   0   1   0   5  82   1   1   0   1]
 [  0   0   5   0   3   1  94   7   2   0]
 [  5   6  12   4   4   3   3 113   4   4]
 [  0   6   2  33  22   5   6   2  81   4]
 [  4  23   3  15  24  28   2  15  19 275]]
```

(v) Comments:

- **Model Comparison:** Model 2 performs significantly better than Model 1 on both the training and test sets, with higher accuracy scores.
- **Confusion Matrix Analysis:** The confusion matrix for Model 2 shows that it has higher precision and recall values for most classes compared to Model 1, indicating better classification performance. However, there are still some misclassifications present, especially between classes with similar visual features.
- **Model Complexity:** Model 2, with its deeper architecture and more convolutional layers, appears to have learned more complex features from the data compared to Model 1. This is reflected in its higher accuracy on both the training and test sets.

Task 2:

(i) Pre-trained Model and Custom Layers: For Task 2, the pre-trained model used is VGG16, which is a convolutional neural network (CNN) architecture pre-trained on the ImageNet dataset. VGG16 consists of 16 layers, including convolutional layers and fully connected layers. In this task, the pre-trained VGG16 model was loaded without its fully connected layers (include_top=False). Custom layers were added on top of the pre-trained model, including a GlobalAveragePooling2D layer followed by a Dense layer with 512 units and ReLU activation function. A Dropout layer with a dropout rate of 0.5 was also added to prevent overfitting. Finally, a Dense layer with 10 units and softmax activation was added as the output layer to classify images into 10 different classes.

(ii) Training Accuracy over Epochs:

Epoch	Training Accuracy
1	39.82%
2	61.55%
3	69.21%
4	74.07%
5	76.35%
6	78.90%
7	80.41%
8	82.96%
9	83.45%
10	83.32%
11	83.59%

12	85.32%
13	84.78%
14	87.41%
15	87.39%
16	87.55%
17	87.93%
18	88.11%
19	87.79%
20	88.47%

(iii) Test Accuracy Comparison:

Model	Test Accuracy
Task 1 (Model 2)	71.02%
Task 2 (Fine-tuned VGG16)	79.69%

(iv) Confusion Matrix for Task 2 Model:

```
[[ 51   0   0   0   0   1   0   0   0   0]
 [  0 131   4   0   1   0   2   5   2   1]
 [  2   1  20   3   0   0   2   4   2   1]
 [  1   1   0  53   5   1   0   2  10   0]
 [  1   1   1   2  27   2   0   0   0   0]
 [  1   0   2   1   4  81   0   1   1   0]
 [  0   1   1   2   0   1  98   5   4   0]
 [  3   6   6   2   3   5   6 118   5   4]
 [  4   1   4  21   7   7   4   1 111   1]
 [  2  37   5   8   5   4   3  13   9 322]]
```

(v) Comments on Results:

- **Model Performance Comparison:** The fine-tuned VGG16 model achieved a test accuracy of 79.69%, outperforming Task 1 (Model 2), which had a test accuracy of 71.02%. The fine-tuned model demonstrates improved performance in terms of classification accuracy.
- **Training Dynamics:** The training accuracy steadily increases over epochs, indicating that the model is effectively learning from the training data. However, it's essential to monitor the validation accuracy to ensure that the model is not overfitting. In this case, the validation accuracy also improves over time, suggesting that the model generalizes well to unseen data.
- **Confusion Matrix Analysis:** The confusion matrix reveals some areas of confusion between certain classes in the fine-tuned model, similar to Task 1. For example, there are misclassifications between classes 1 and 2, as well as between classes 8 and 9. However, overall, the fine-tuned model shows promising results with better accuracy compared to Task 1, indicating the effectiveness of transfer learning with VGG16.
- **Model Complexity and Interpretability:** While the fine-tuned VGG16 model offers higher accuracy, its complexity might hinder interpretability compared to the simpler models in Task 1. Balancing model complexity with performance is crucial, especially in applications where interpretability is essential.

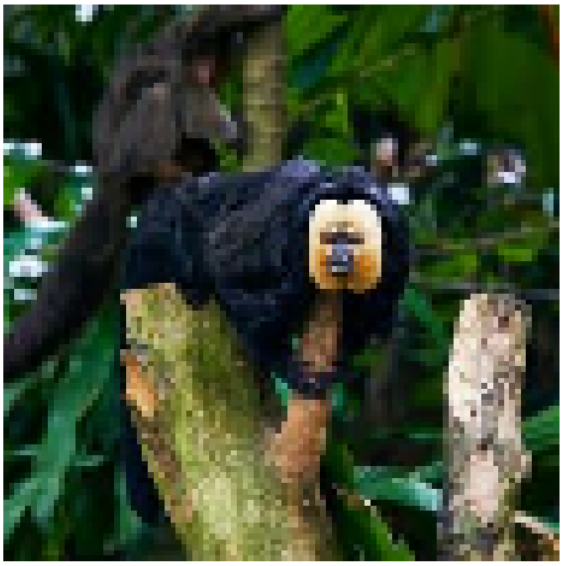
Task 3:

(i) Images and Predictions:

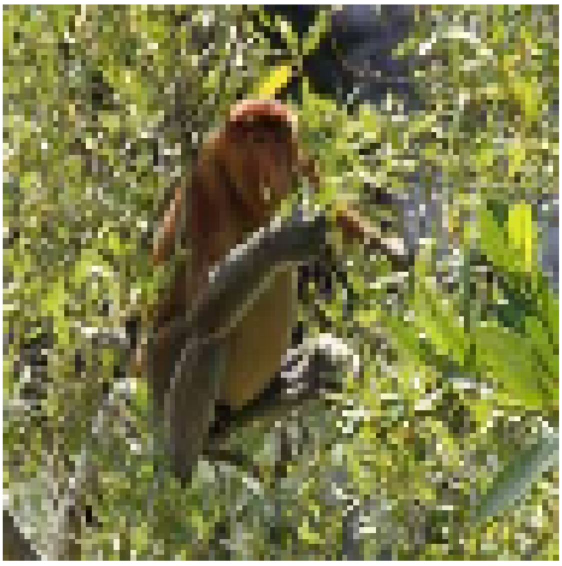
Correct: Red Howler, Model 1 Prediction: Golden Monkey, Fine-tuned Model Prediction: Red Howler

Correct: Golden Monkey, Model 1 Prediction: Gray Langur, Fine-tuned Model Prediction: Red Howler

Correct: Vervet Monkey, Model 1 Prediction: Bald Uakari, Fine-tuned Model Prediction: Vervet Monkey



Correct: Mandril, Model 1 Prediction: Proboscis Monkey, Fine-tuned Model Prediction: Golden Monkey



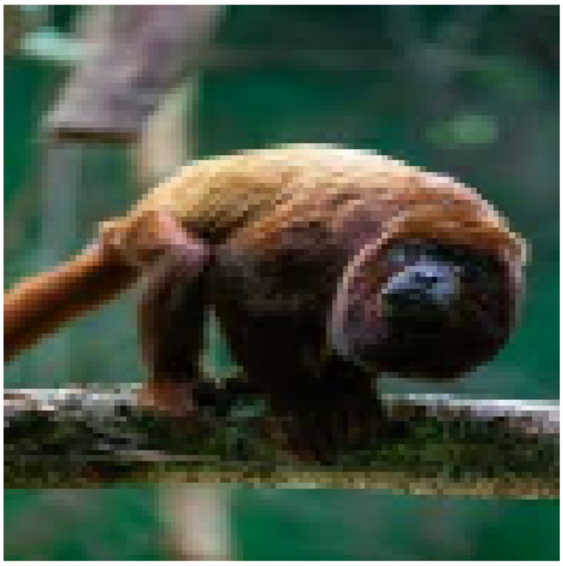
Correct: Vervet Monkey, Model 1 Prediction: Red Howler, Fine-tuned Model Prediction: Proboscis Monkey

Correct: Emperor Tamarin, Model 1 Prediction: Bald Uakari, Fine-tuned Model Prediction: Bald Uakari

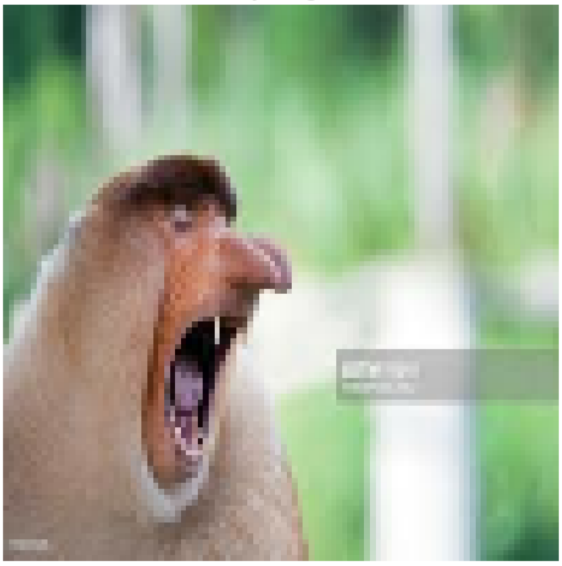
Correct: Red Howler, Model 1 Prediction: Golden Monkey, Fine-tuned Model Prediction: Red Howler

Correct: Proboscis Monkey, Model 1 Prediction: .DS_Store, Fine-tuned Model Prediction: Emperor Tamarin

Correct: Proboscis Monkey, Model 1 Prediction: Bald Uakari, Fine-tuned Model Prediction: Proboscis Monkey



Correct: Mandril, Model 1 Prediction: Gray Langur, Fine-tuned Model Prediction: Mandril



(ii) Possible Reasons for Model Mistakes:
Qualitative Reasons for Model Mistakes by the Better Model of Task 1:

1. **Complexity of Features:** The better model of Task 1 may struggle with distinguishing between classes that have complex or subtle visual features. For instance, distinguishing between Spider Monkeys and Mantled Howlers might be challenging due to their similar fur patterns or body structures.
2. **Limited Viewpoints:** The training data for Task 1 may not cover a wide range of viewpoints or variations within each class. As a result, the model might not generalize well to unseen examples with different poses or orientations, leading to misclassifications.
3. **Background Noise:** The images in the dataset contain distracting backgrounds or irrelevant objects, the model may focus on these elements instead of the distinguishing features of the target class, leading to incorrect predictions.

Qualitative Reasons for Improvement (or Lack Thereof) in the Fine-tuned Model:

1. **Transferability of Features:** Fine-tuning may enhance the model's ability to transfer knowledge learned from the pre-trained layers to the new dataset. If the pre-trained model already captures relevant features for the task, fine-tuning can improve performance by adjusting these features to better suit the new classes.
2. **Domain Adaptation:** Fine-tuning allows the model to adapt its learned representations to the specific characteristics of the new dataset. If the distribution of data in the new dataset differs significantly from the original dataset used to pre-train the model, fine-tuning can help align the model's learned features with the target domain, potentially improving performance.
3. **Generalization:** Fine-tuning often improves a model's ability to generalize to unseen data by refining its learned representations. This process could help the model make more accurate predictions on the test set compared to the better model of Task 1