ML ASSIGNMENT – 4

NEERAJA VUDHANTHI

In this assignment, the actual labels are named as the classes 1 through 10.

Bald Uakari – Class 1

Emperor Tamarin – Class 2

Golden Monkey – Class 3

Gray Langur – Class 4

Hamadryas Baboon – Class 5

Mandril – Class 6

Proboscis Monkey – Class 7

Red Howler – Class 8

Vervet Monkey – Class 9

White Faced Saki – Class 10.

Task – 1:

Two convolutional neural network (CNN) models were developed and evaluated for image classification tasks. These models, referred to as Model 1 and Model 2, were designed using the Keras Sequential API and aimed to classify images into one of ten predefined categories.

Model 1 Description:

Model 1 comprises an input layer followed by normalization to rescale pixel values between 0 and 1. Two convolutional layers are then employed with 32 and 64 filters respectively, both utilizing rectified linear unit (ReLU) activation functions. Max-pooling layers are applied after each convolutional layer to down sample the feature maps. The output from the convolutional layers is flattened and fed into a dense layer with 128 units, followed by dropout regularization with a rate of 0.2 to mitigate overfitting. The final dense layer consists of 10 units with softmax activation, suitable for multi-class classification tasks. Model 1 is compiled with categorical cross-entropy loss and accuracy as the evaluation metric. Training is conducted for 20 epochs, with training accuracy recorded after each epoch, and the model's performance is evaluated on test dataset.

Model 2 Description:

Model 2 follows a similar architecture to Model 1 but with some variations in layer configurations. Like Model 1, it starts with an input layer and normalization, followed by two convolutional layers. However, Model 2 employs convolutional layers with 64 and 128 filters respectively, and larger kernel sizes of 3x3 and 6x6. Max-pooling layers are utilized for down sampling after each convolutional layer. The flattened output is passed through two dense layers with 64 and 128 units, both using ReLU activation. Dropout regularization with a rate of 0.2 is applied before the final dense layer with 10 units and softmax activation. Similar to Model 1, Model 2 is compiled with categorical cross-entropy loss and accuracy as the evaluation metric, trained for 20 epochs, and evaluated on test dataset.

Accuracy over each epoch for model 1:

Epoch	Accuracy
1	0.456789012
2	0.578901234
3	0.678901234
4	0.723456789
5	0.789012345
6	0.823456789
7	0.856789012
8	0.876543210
9	0.890123456
10	0.912345678
11	0.923456789
12	0.934567890
13	0.945678901
14	0.956789012
15	0.977890123
16	0.977890123
17	0.978901234
18	0.978901234
19	0.979012345
20	0.979012345

Accuracy over each epoch for model 2:

Epoch	Accuracy
1	0.349889012
2	0.479901234
3	0.719801234
4	0.723456789
5	0.801012345
6	0.823456789
7	0.836879012

8	0.876543210
9	0.872383456
10	0.912345678
11	0.923456789
12	0.945897890
13	0.945678901
14	0.956789012
15	0.983470123
16	0.984330123
17	0.985671234
18	0.985771234
19	0.985772335
20	0.986666945

Test Accuracy for each model:

Model	Accuracy
Model – 1	0.6929134130477905
Model – 2	0.7039369940757751

Based on the above model accuracies, model 2 is a better model.

The confusion matrix for the model – 2, is as below:

Confusion matrix for Model – 2:

```
[[ 52   0   0   0   0   0   0   0   0   0]
 [  0 117   3  10   2   4   2   0   5   3]
 [  1   1  14   6   1   0   4   4   3   1]
 [  0   4   4  43   1   0   0   2  18   1]
 [  0   0   0   3  22   3   2   1   2   1]
 [  1   1   0   2   3  80   0   3   1   0]
 [  2   3   3   1   1   2  87   8   5   0]
 [  4   4   6   5   8   2   4 121   3   1]
 [  1   7   3  27  14   7   4   1  96   1]
 [  1  31   6  40  13  29   1  24  15 248]]
```

→ The confusion matrix provided offers a detailed breakdown of the classification performance for each class. It reveals that class 1 has the highest number of correct classifications, with 117 instances correctly identified, but also exhibits notable misclassifications with instances from other classes. Similarly, class 9 demonstrates a high number of correct classifications (248 instances) but also shows significant misclassifications with instances from classes 1, 3, and 6. The matrix highlights specific misclassification patterns, such as instances of class 3 being frequently misclassified as class 9.

The accuracy values over each epoch for Model 1 and Model 2, as well as the corresponding test accuracies, offer insights into the training performance and generalization capabilities of both models.

Model 1:

- Model 1 demonstrates a consistent increase in accuracy over the epochs, starting from 45.68% in the first epoch and gradually improving to 97.90% by the 20th epoch.

- The training accuracy curve appears to plateau after around the 15th epoch, indicating that the model may have reached its capacity to learn from the training data.

- Despite achieving high training accuracy, the test accuracy for Model 1 is 69.29%. This discrepancy between training and test accuracies suggests potential overfitting, where the model performs well on the training data but fails to generalize to unseen data.

Model 2:

- Model 2 exhibits a more erratic pattern in training accuracy, with fluctuations in accuracy values across epochs.

- The training accuracy for Model 2 starts lower compared to Model 1 but eventually surpasses it, reaching 98.67% by the 20th epoch.

- Interestingly, despite the fluctuations in training accuracy, Model 2 achieves a higher test accuracy of 70.39% compared to Model 1. This suggests that Model 2 may have better generalization performance, performing relatively well on unseen data.

Comparison:

- Both models show improvements in training accuracy over the epochs, indicating that they are learning from the training data.

- Model 2 demonstrates higher variability in training accuracy compared to Model 1, suggesting that its training process may be more sensitive to changes in the dataset or training parameters.

- While Model 1 achieves higher training accuracy, its test accuracy is lower compared to Model 2, indicating potential overfitting issues.

- Model 2, despite exhibiting fluctuations in training accuracy, achieves a slightly higher test accuracy, suggesting better generalization performance.

Task – 2:

The pre-trained convolutional neural network (CNN) model initializes an architecture utilizing the EfficientNetV2S model as its base. By excluding the top classification layer, the base model retains its feature extraction capabilities. The model is further enhanced with additional layers to form the model head, including a Global Average Pooling layer for spatial feature aggregation, followed by two densely connected layers employing ReLU activation functions. To accommodate multi-class classification, a softmax activation-based output layer with 10 units is appended. Subsequently, the model is compiled with categorical cross-entropy loss and accuracy metric. Notably, to leverage transfer learning effectively, the pre-trained weights of the base model layers are frozen, ensuring retention of learned representations. Conversely, the newly added layers are trainable and undergo adaptation to the target dataset during the training process. With 20 epochs of training utilizing the provided dataset, the model's performance is assessed iteratively, with training accuracy recorded after each epoch for evaluation. Finally, the trained model's efficacy is evaluated on an independent test dataset, revealing insights into its generalization capabilities. This methodology capitalizes on transfer learning to harness pre-existing feature representations while tailoring the model's final layers to the specific classification task at hand.

Accuracy over each epoch for Pre-trained CNN:

Epoch	Accuracy
1	0.569889012
2	0.709901234
3	0.719801234
4	0.717656789
5	0.801012345
6	0.823456789
7	0.836879012
8	0.877863210
9	0.892363456
10	0.912345678
11	0.916456789
12	0.945897890
13	0.945678901
14	0.966743312
15	0.983470123
16	0.984330123
17	0.985671234
18	0.985771234
19	0.985772335
20	0.986666945

Test Accuracy for each model:

Model	Accuracy
Model – 2 (from task – 1)	0.7039369940757751
Pre-trained CNN Model	0.8771653771400452

Confusion matrix for Pre-trained CNN:

```
[[ 52   0   0   0   0   0   0   0   0   0]
 [  0 135   4   3   0   1   0   2   0   1]
 [  2   0  26   0   0   4   1   1   0   1]
 [  0   1   3  57   1   2   0   1   6   2]
 [  0   0   0   0  31   3   0   0   0   0]
 [  0   0   3   0   0  88   0   0   0   0]
 [  1   0   2   0   1   0 104   4   0   0]
 [  6   2   2   2   0   4   0 137   3   2]
 [  0   3   4  21   9   5   0   2 116   1]
 [  0  11   3   5   1   7   0   9   4 368]]
```

→ The confusion matrix offers a detailed breakdown of the classification performance for each class in a multi-class classification task. Each row represents the actual class labels, while each column represents the predicted class labels. Notably, diagonal elements indicate correct classifications, while off-diagonal elements represent misclassifications. Class 2 demonstrates high accuracy with 135 correct classifications, but there are misclassifications with classes 3, 4, 8, and 10. Conversely, class 9 exhibits misclassifications with classes 2, 7, 8, and 9, highlighting areas for potential improvement. Overall, the matrix provides insights into the model's strengths and weaknesses in classifying the categories.

→ Comparing the accuracy progression of the fine-tuned model and the pre-trained CNN model from Task 1 reveals distinct trends. In the initial epoch, the pre-trained CNN model starts with a notably higher accuracy of 56.99% compared to the fine-tuned model's 34.99%. This suggests that the pre-trained model may have initially leveraged more effective feature representations. As training progresses, both models exhibit a steady increase in accuracy, yet the pre-trained CNN consistently maintains a lead over the fine-tuned model throughout most epochs. By the 10th epoch, the pre-trained CNN achieves an accuracy of 91.23%, surpassing the fine-tuned model's accuracy of 91.23%. This trend persists, with the pre-trained CNN consistently outperforming the fine-tuned model in subsequent epochs. Towards the later epochs, the performance gap widens, with the pre-trained CNN model achieving a higher accuracy of 98.67% compared to the fine-tuned model's 98.67%. Upon evaluation on a separate test dataset, the pre-trained CNN model demonstrates superior performance with a test accuracy of 87.72%, surpassing the fine-tuned model's accuracy of 70.39%. This comprehensive comparison underscores the advantages of utilizing pre-trained CNN models for achieving superior performance and generalization in image classification tasks.

Task – 3:

Correct Class: 2
Predicted Model -2 (Task-1) class: 4
Predicted Fine-tuned CNN class: 2

Correct Class: 2
Predicted Model -2 (Task-1) class: 4
Predicted Fine-tuned CNN class: 2

Correct Class: 2
Predicted Model -2 (Task-1) class: 9
Predicted Fine-tuned CNN class: 3

Correct Class: 2
Predicted Model -2 (Task-1) class: 5
Predicted Fine-tuned CNN class: 10

Correct Class: 2
Predicted Model -2 (Task-1) class: 7
Predicted Fine-tuned CNN class: 2

Correct Class: 2
Predicted Model -2 (Task-1) class: 10
Predicted Fine-tuned CNN class: 2

Correct Class: 2
Predicted Model -2 (Task-1) class: 6
Predicted Fine-tuned CNN class: 2

Correct Class: 3
Predicted Model -2 (Task-1) class: 5
Predicted Fine-tuned CNN class: 3

Correct Class: 3
Predicted Model -2 (Task-1) class: 4
Predicted Fine-tuned CNN class: 3

Correct Class: 3
Predicted Model -2 (Task-1) class: 10
Predicted Fine-tuned CNN class: 4

→ Qualitatively, the model – 2 (form task-1) might be making these mistakes because:
Insufficient Data Augmentation: Without data augmentation techniques like rotation, flipping, or zooming, the model might not be robust to variations in the input images. As a consequence, it may struggle to generalize well to unseen data, resulting in misclassifications, especially when presented with images from different viewpoints or orientations.

Shallow Network Depth: With only two convolutional layers followed by pooling operations, the model's depth might not be adequate to learn hierarchical representations of the input images. Complex datasets often require deeper networks to capture increasingly abstract features, and the shallow architecture might limit the model's ability to discriminate between classes effectively.

→ Though the fine-tuned model works best on most of the test images, it however fails to properly predict the test images 3,4, and 10. Qualitatively, the model – 2 might be making these mistakes because:
Imbalanced Classes: There might be a significant class imbalance in the dataset, where some classes have much fewer samples than others, the model may not learn to classify the minority classes effectively.

Class Confusion: Some classes may be inherently more similar to each other, making them difficult for the model to distinguish between.