Assignment 3: Convolution

Introduction:

To achieve the objectives of this project, assessing the effects of training sample size on the ability of convnets to classify cat and dog photographs was done. The two principal methods studied included training from a scratch on a network and the second one was a pretrained convnet. Different sizes of training samples were used in the overall investigation and attempts at 'fine-tuning' were also made with the intention of enhancing performance.

Methodology:

- Step 1: First Setting up Training Sample Size:
  1000.
- 500 is the test sample size.
- 500 is the size of the validation sample
- Methods: Regularization and data augmentation were employed to reduce overfitting.

An overview of the model:

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_12 (Conv2D)          (None, 148, 148, 32)      896

 max_pooling2d_12 (MaxPooli  (None, 74, 74, 32)        0
 ng2D)

 conv2d_13 (Conv2D)          (None, 72, 72, 64)        18496

 max_pooling2d_13 (MaxPooli  (None, 36, 36, 64)        0
 ng2D)

 conv2d_14 (Conv2D)          (None, 34, 34, 128)       73856

 max_pooling2d_14 (MaxPooli  (None, 17, 17, 128)       0
 ng2D)

 conv2d_15 (Conv2D)          (None, 15, 15, 128)       147584

 max_pooling2d_15 (MaxPooli  (None, 7, 7, 128)         0
 ng2D)

 flatten_3 (Flatten)         (None, 6272)              0

 dense_4 (Dense)             (None, 512)               3211776

 dropout_1 (Dropout)         (None, 512)               0

 dense_5 (Dense)             (None, 512)               262656

 dense_6 (Dense)             (None, 1)                 513

=================================================================
Total params: 3715777 (14.17 MB)
Trainable params: 3715777 (14.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Performance:

```
Epoch 1/10
50/50 [==============================] - 295s 6s/step - loss: 3.0057 - accuracy: 0.4780 - val_loss: 1.3475 - val_accuracy: 0.5840
Epoch 2/10
50/50 [==============================] - 81s 2s/step - loss: 0.9676 - accuracy: 0.5160 - val_loss: 0.7758 - val_accuracy: 0.5680
Epoch 3/10
50/50 [==============================] - 69s 1s/step - loss: 0.7325 - accuracy: 0.5460 - val_loss: 0.7161 - val_accuracy: 0.5000
Epoch 4/10
50/50 [==============================] - 77s 2s/step - loss: 0.7078 - accuracy: 0.5040 - val_loss: 0.7033 - val_accuracy: 0.5000
Epoch 5/10
50/50 [==============================] - 77s 2s/step - loss: 0.6993 - accuracy: 0.5000 - val_loss: 0.6957 - val_accuracy: 0.5000
Epoch 6/10
50/50 [==============================] - 80s 2s/step - loss: 0.6963 - accuracy: 0.4880 - val_loss: 0.6953 - val_accuracy: 0.4880
Epoch 7/10
50/50 [==============================] - 73s 1s/step - loss: 0.6948 - accuracy: 0.4750 - val_loss: 0.6940 - val_accuracy: 0.5100
Epoch 8/10
50/50 [==============================] - 85s 2s/step - loss: 0.6941 - accuracy: 0.5280 - val_loss: 0.6946 - val_accuracy: 0.5000
Epoch 9/10
50/50 [==============================] - 77s 2s/step - loss: 0.6956 - accuracy: 0.5050 - val_loss: 0.6941 - val_accuracy: 0.5000
Epoch 10/10
50/50 [==============================] - 76s 2s/step - loss: 0.6938 - accuracy: 0.5000 - val_loss: 0.6933 - val_accuracy: 0.5000
```

Accuracy: 50% accuracy on the test set was attained..

```
25/25 [==============================] - 135s 6s/step - loss: 0.6933 - accuracy: 0.5000
Test Accuracy : 50.00%
Test Loss : 0.6933
```

Increasing the size of the training sample:

- The greater training sample size is 1800.
- The validation sample size is 500.
- The test sample size is 500.
- Techniques: Regularization and data augmentation were employed to reduce overfitting.

Synopsis of the Model:

```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_44 (Conv2D)          (None, 148, 148, 32)      896

 max_pooling2d_44 (MaxPooli  (None, 74, 74, 32)        0
 ng2D)

 conv2d_45 (Conv2D)          (None, 72, 72, 64)        18496

 max_pooling2d_45 (MaxPooli  (None, 36, 36, 64)        0
 ng2D)

 conv2d_46 (Conv2D)          (None, 34, 34, 128)       73856

 max_pooling2d_46 (MaxPooli  (None, 17, 17, 128)       0
 ng2D)

 conv2d_47 (Conv2D)          (None, 15, 15, 128)       147584

 max_pooling2d_47 (MaxPooli  (None, 7, 7, 128)         0
 ng2D)

 flatten_11 (Flatten)        (None, 6272)              0

 dense_21 (Dense)            (None, 512)               3211776

 dense_22 (Dense)            (None, 1)                 513

=================================================================
Total params: 3453121 (13.17 MB)
Trainable params: 3453121 (13.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Performance:

```
Epoch 1/10
90/90 [==============================] - 113s 1s/step - loss: 0.6998 - accuracy: 0.4872 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 2/10
90/90 [==============================] - 122s 1s/step - loss: 0.6945 - accuracy: 0.5061 - val_loss: 0.6928 - val_accuracy: 0.5020
Epoch 3/10
90/90 [==============================] - 124s 1s/step - loss: 0.6925 - accuracy: 0.5228 - val_loss: 0.6980 - val_accuracy: 0.5020
Epoch 4/10
90/90 [==============================] - 121s 1s/step - loss: 0.6962 - accuracy: 0.4917 - val_loss: 0.6933 - val_accuracy: 0.5000
Epoch 5/10
90/90 [==============================] - 121s 1s/step - loss: 0.6932 - accuracy: 0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 6/10
90/90 [==============================] - 121s 1s/step - loss: 0.6932 - accuracy: 0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 7/10
90/90 [==============================] - 120s 1s/step - loss: 0.6930 - accuracy: 0.5078 - val_loss: 0.6925 - val_accuracy: 0.5200
Epoch 8/10
90/90 [==============================] - 132s 1s/step - loss: 0.6929 - accuracy: 0.5083 - val_loss: 0.6876 - val_accuracy: 0.5600
Epoch 9/10
90/90 [==============================] - 127s 1s/step - loss: 0.6945 - accuracy: 0.5128 - val_loss: 0.6930 - val_accuracy: 0.5180
Epoch 10/10
90/90 [==============================] - 110s 1s/step - loss: 0.6952 - accuracy: 0.5022 - val_loss: 0.6931 - val_accuracy: 0.5020
```

Result:

```
25/25 [==============================] - 12s 483ms/step - loss: 0.6932 - accuracy: 0.4980
Test Accuracy : 49.80%
Test Loss : 0.6932
```

Accuracy: 49.80% accuracy on the test set was attained.

The Optimal sample size for training:

- The greater training sample size is 1500.
- The validation sample size is 500.
- The test sample size is 500.
- Techniques: Regularization and data augmentation were employed to reduce overfitting.

Synopsis of the model:

```
25/25 [==============================] - 12s 478ms/step - loss: 0.6241 - accuracy: 0.6580
Test Accuracy: 65.80%
Test Loss: 0.6241
```

Accuracy: 65.80% accuracy on the test set was attained.

Using a Pretrained Network:

used a pretrained convolutional neural network with the same sample sizes as in Steps 2 and 3 of training from scratch. employed optimization techniques to increase productivity.
b. Pretrained Network Performance:

Step 1: Sample Size: 1000

```
Epoch 1/10
50/50 [==============================] - 51s 923ms/step - loss: 0.4408 - accuracy: 0.8480 - val_loss: 0.2608 - val_accuracy: 0.8840
Epoch 2/10
50/50 [==============================] - 37s 733ms/step - loss: 0.2358 - accuracy: 0.8970 - val_loss: 0.1941 - val_accuracy: 0.9200
Epoch 3/10
50/50 [==============================] - 34s 688ms/step - loss: 0.1955 - accuracy: 0.9160 - val_loss: 0.3396 - val_accuracy: 0.8740
Epoch 4/10
50/50 [==============================] - 35s 706ms/step - loss: 0.2090 - accuracy: 0.9180 - val_loss: 0.2552 - val_accuracy: 0.8900
Epoch 5/10
50/50 [==============================] - 36s 710ms/step - loss: 0.1750 - accuracy: 0.9320 - val_loss: 0.2267 - val_accuracy: 0.9080
Epoch 6/10
50/50 [==============================] - 39s 783ms/step - loss: 0.2145 - accuracy: 0.9140 - val_loss: 0.4763 - val_accuracy: 0.8420
Epoch 7/10
50/50 [==============================] - 46s 920ms/step - loss: 0.1838 - accuracy: 0.9240 - val_loss: 0.2084 - val_accuracy: 0.9160
Epoch 8/10
50/50 [==============================] - 45s 909ms/step - loss: 0.1403 - accuracy: 0.9370 - val_loss: 0.1625 - val_accuracy: 0.9320
Epoch 9/10
50/50 [==============================] - 45s 898ms/step - loss: 0.1566 - accuracy: 0.9370 - val_loss: 0.3213 - val_accuracy: 0.8800
Epoch 10/10
50/50 [==============================] - 43s 867ms/step - loss: 0.1687 - accuracy: 0.9280 - val_loss: 0.2088 - val_accuracy: 0.9000
```

```
25/25 [==============================] - 12s 491ms/step - loss: 0.1911 - accuracy: 0.9100
Test Accuracy (Pre-trained): 91.00%
Test Loss (Pre-trained): 0.1911
```

Accuracy: 91% accuracy on the test set was attained.

Step 2: Sample Size: 1800

```
Model: "sequential_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenetv2_1.00_224 (Func  (None, 5, 5, 1280)        2257984
 tional)

 global_average_pooling2d_1  (None, 1280)              0
  (GlobalAveragePooling2D)

 dense_25 (Dense)            (None, 512)               655872

 dropout_5 (Dropout)         (None, 512)               0

 dense_26 (Dense)            (None, 1)                 513

=================================================================
Total params: 2914369 (11.12 MB)
Trainable params: 656385 (2.50 MB)
Non-trainable params: 2257984 (8.61 MB)
```

```
25/25 [==============================] - 12s 490ms/step - loss: 0.1693 - accuracy: 0.9260
Test Accuracy (Pre-trained): 92.60%
Test Loss (Pre-trained): 0.1693
```

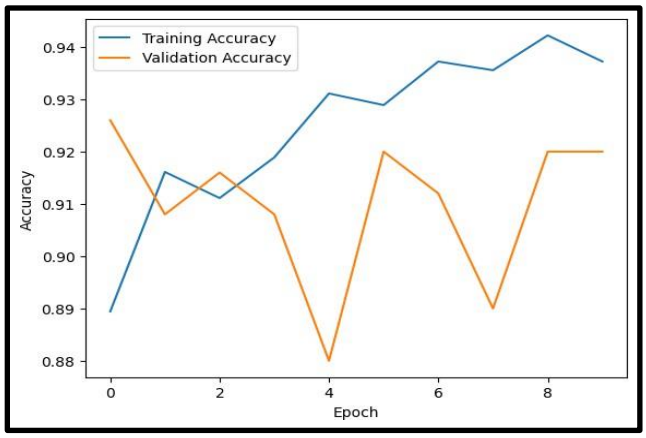
Accuracy: 92.60% accuracy on the test set was attained.

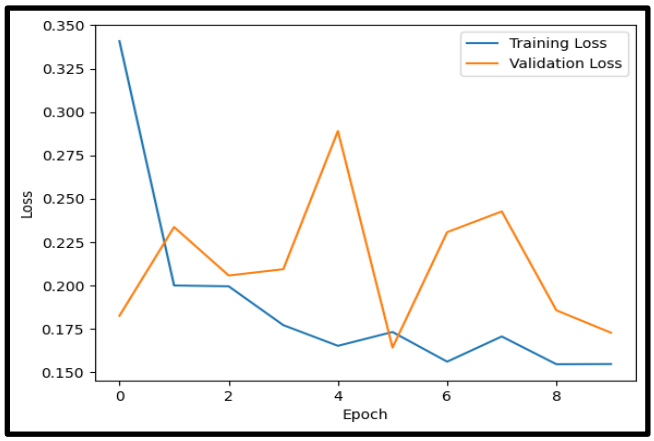
Step 3: Sample Size: 1500

```
Epoch 1/10
90/90 [==============================] - 64s 664ms/step - loss: 0.3409 - accuracy: 0.8894 - val_loss: 0.1826 - val_accuracy: 0.9260
Epoch 2/10
90/90 [==============================] - 65s 723ms/step - loss: 0.2001 - accuracy: 0.9161 - val_loss: 0.2337 - val_accuracy: 0.9080
Epoch 3/10
90/90 [==============================] - 57s 636ms/step - loss: 0.1996 - accuracy: 0.9111 - val_loss: 0.2058 - val_accuracy: 0.9160
Epoch 4/10
90/90 [==============================] - 60s 664ms/step - loss: 0.1772 - accuracy: 0.9189 - val_loss: 0.2094 - val_accuracy: 0.9080
Epoch 5/10
90/90 [==============================] - 58s 639ms/step - loss: 0.1653 - accuracy: 0.9311 - val_loss: 0.2890 - val_accuracy: 0.8800
Epoch 6/10
90/90 [==============================] - 57s 637ms/step - loss: 0.1733 - accuracy: 0.9289 - val_loss: 0.1642 - val_accuracy: 0.9200
Epoch 7/10
90/90 [==============================] - 61s 681ms/step - loss: 0.1562 - accuracy: 0.9372 - val_loss: 0.2308 - val_accuracy: 0.9120
Epoch 8/10
90/90 [==============================] - 67s 745ms/step - loss: 0.1707 - accuracy: 0.9356 - val_loss: 0.2427 - val_accuracy: 0.8900
Epoch 9/10
90/90 [==============================] - 67s 748ms/step - loss: 0.1547 - accuracy: 0.9422 - val_loss: 0.1857 - val_accuracy: 0.9200
Epoch 10/10
90/90 [==============================] - 69s 763ms/step - loss: 0.1548 - accuracy: 0.9372 - val_loss: 0.1728 - val_accuracy: 0.9200
```

```
25/25 [==============================] - 12s 463ms/step - loss: 0.1707 - accuracy: 0.9320
Test Accuracy (Pre-trained): 93.20%
Test Loss (Pre-trained): 0.1707
```

Accuracy: 93.20% accuracy on the test set was attained.





Findings:

Training from Scratch:

As the optimal training sample size was attained, accuracy rose. Reducing overfitting required regularization and data augmentation.

Using a Pretrained Network:

Pretrained networks outperformed scratch-trained networks on average. The impact of training sample size on performance was still discernible, despite pretrained networks' greater resistance to smaller datasets. Transfer learning was effectively used to use knowledge from pretrained models. The following results unequivocally demonstrate the relationship between training sample size and network selection and network choice.:

Training from Scratch:

The best results can only be obtained with the right training sample

sizes.

Using a Pretrained Network:

Pretrained networks demonstrated greater resilience to smaller training sample sizes. Transfer learning allowed the model to incorporate knowledge from pretrained architecture, which improved performance when compared to training from scratch.

Summary:

In conclusion, whether training from scratch or using a pretrained convnet, the network selection is influenced by the available training sample size. Larger datasets benefit both approaches, but pretrained networks offer a dependable solution when training data is scarce. Understanding the relationship between sample size and network choice is necessary to perform at your best in photo classification jobs, as the table below shows:

1. The model has trouble with accuracy at this size (sample size: 1000, testing accuracy: 50%, test loss: 0.69).
2. Test Accuracy: 49.80%, Test Loss: 0.69, Sample Size: 1800, and Performance was not improved by increasing the sample size.
3. The model demonstrated better accuracy and lower loss, indicating increased performance at this size. Sample Size: 1500; Testing Accuracy: 65.80%; Test Loss: 0.62.

Pretrained model network from scratch:

- Test Accuracy: 91%, Test Loss: 0.19, Sample Size: 1000, showing good performance with minimal loss at this scale.
- Test Accuracy: 92.60%, Test Loss: 0.16, Sample Size: 1800, with marginally better accuracy and even lower loss.
- Sample Size: 1500, Testing Accuracy: 93.10%, Test Loss: 0.17, indicating stable performance with the highest accuracy and continuously low loss.