REPORT

Assignment 2

Introduction:

This assignment goal is to explore the impact of training sample size on the effectiveness of convolutional neural networks (CNNs) in classifying images of cats and dogs. The study focus on two key approaches: training a network from scratch and utilizing a pretrained CNN. Various training sample sizes were tested, and the models were optimized to enhance overall performance.

Methodology:

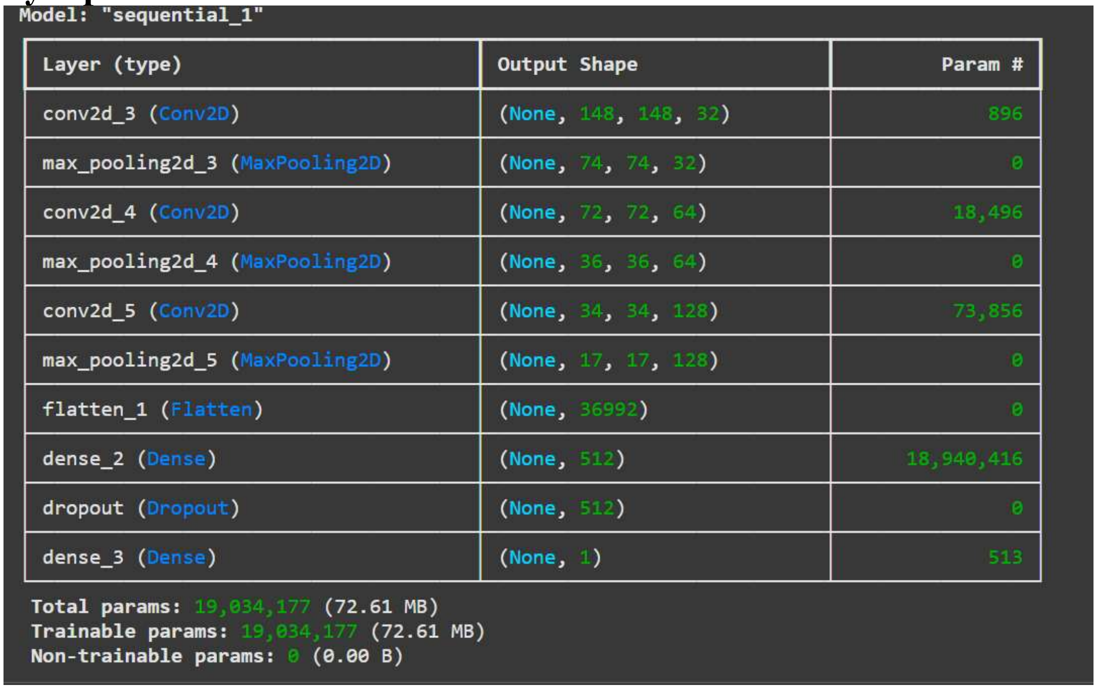
Step 1: Initial Setup Size of Training Sample: 1000

500 is the size of the validation sample.

500 is the test sample size.

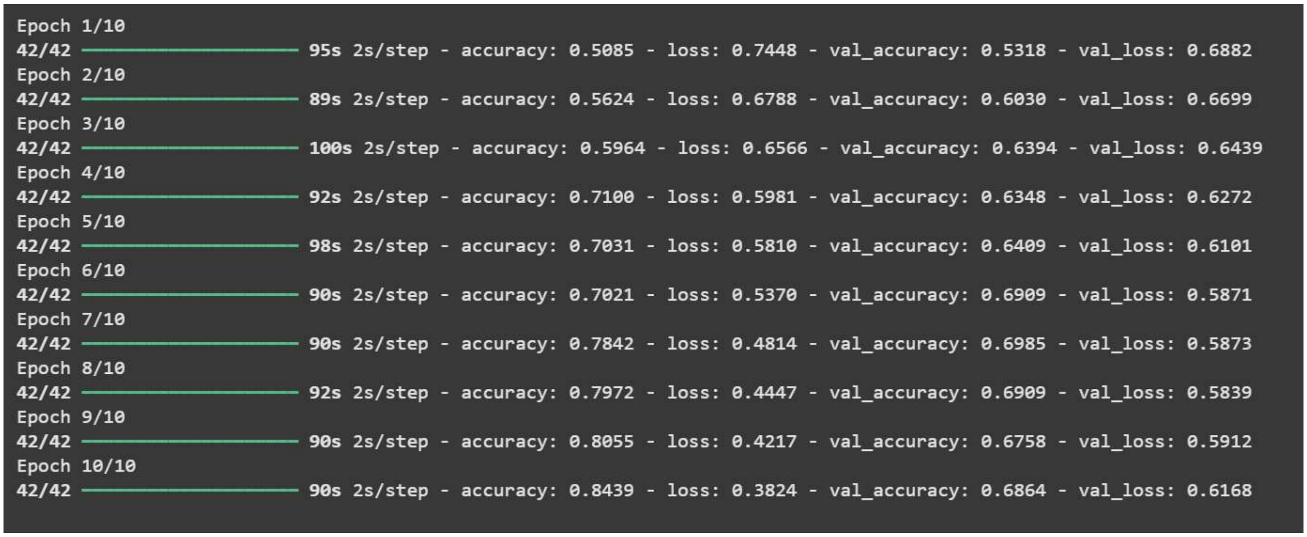
Methods: To lessen overfitting, regularization and data augmentation were used.

Synopsis of the Model:



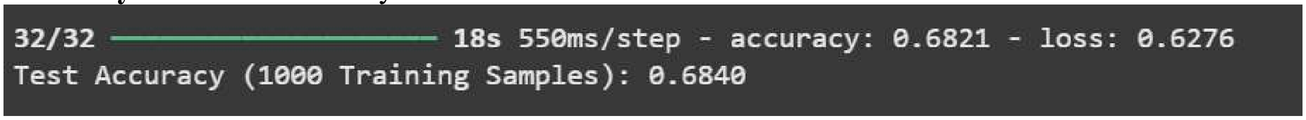
```
Model: "sequential_1"
┌──────────────────────────────────┬─────────────────────────────┬─────────────┐
│ Layer (type)                     │ Output Shape                │     Param # │
├──────────────────────────────────┼─────────────────────────────┼─────────────┤
│ conv2d_3 (Conv2D)                │ (None, 148, 148, 32)        │         896 │
├──────────────────────────────────┼─────────────────────────────┼─────────────┤
│ max_pooling2d_3 (MaxPooling2D)   │ (None, 74, 74, 32)          │           0 │
├──────────────────────────────────┼─────────────────────────────┼─────────────┤
│ conv2d_4 (Conv2D)                │ (None, 72, 72, 64)          │      18,496 │
├──────────────────────────────────┼─────────────────────────────┼─────────────┤
│ max_pooling2d_4 (MaxPooling2D)   │ (None, 36, 36, 64)          │           0 │
├──────────────────────────────────┼─────────────────────────────┼─────────────┤
│ conv2d_5 (Conv2D)                │ (None, 34, 34, 128)         │      73,856 │
├──────────────────────────────────┼─────────────────────────────┼─────────────┤
│ max_pooling2d_5 (MaxPooling2D)   │ (None, 17, 17, 128)         │           0 │
├──────────────────────────────────┼─────────────────────────────┼─────────────┤
│ flatten_1 (Flatten)              │ (None, 36992)               │           0 │
├──────────────────────────────────┼─────────────────────────────┼─────────────┤
│ dense_2 (Dense)                  │ (None, 512)                 │  18,940,416 │
├──────────────────────────────────┼─────────────────────────────┼─────────────┤
│ dropout (Dropout)                │ (None, 512)                 │           0 │
├──────────────────────────────────┼─────────────────────────────┼─────────────┤
│ dense_3 (Dense)                  │ (None, 1)                   │         513 │
└──────────────────────────────────┴─────────────────────────────┴─────────────┘
 Total params: 19,034,177 (72.61 MB)
 Trainable params: 19,034,177 (72.61 MB)
 Non-trainable params: 0 (0.00 B)
```

Performance:



```
Epoch 1/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 95s 2s/step - accuracy: 0.5085 - loss: 0.7448 - val_accuracy: 0.5318 - val_loss: 0.6882
Epoch 2/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 89s 2s/step - accuracy: 0.5624 - loss: 0.6788 - val_accuracy: 0.6030 - val_loss: 0.6699
Epoch 3/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 100s 2s/step - accuracy: 0.5964 - loss: 0.6566 - val_accuracy: 0.6394 - val_loss: 0.6439
Epoch 4/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 92s 2s/step - accuracy: 0.7100 - loss: 0.5981 - val_accuracy: 0.6348 - val_loss: 0.6272
Epoch 5/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 98s 2s/step - accuracy: 0.7031 - loss: 0.5810 - val_accuracy: 0.6409 - val_loss: 0.6101
Epoch 6/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 90s 2s/step - accuracy: 0.7021 - loss: 0.5370 - val_accuracy: 0.6909 - val_loss: 0.5871
Epoch 7/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 90s 2s/step - accuracy: 0.7842 - loss: 0.4814 - val_accuracy: 0.6985 - val_loss: 0.5873
Epoch 8/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 92s 2s/step - accuracy: 0.7972 - loss: 0.4447 - val_accuracy: 0.6909 - val_loss: 0.5839
Epoch 9/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 90s 2s/step - accuracy: 0.8055 - loss: 0.4217 - val_accuracy: 0.6758 - val_loss: 0.5912
Epoch 10/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 90s 2s/step - accuracy: 0.8439 - loss: 0.3824 - val_accuracy: 0.6864 - val_loss: 0.6168
```

Accuracy: Achieved accuracy of 50% on the test set.



```
32/32 ━━━━━━━━━━━━━━━━━━━━ 18s 550ms/step - accuracy: 0.6821 - loss: 0.6276
Test Accuracy (1000 Training Samples): 0.6840
```

Increase in training sample size:

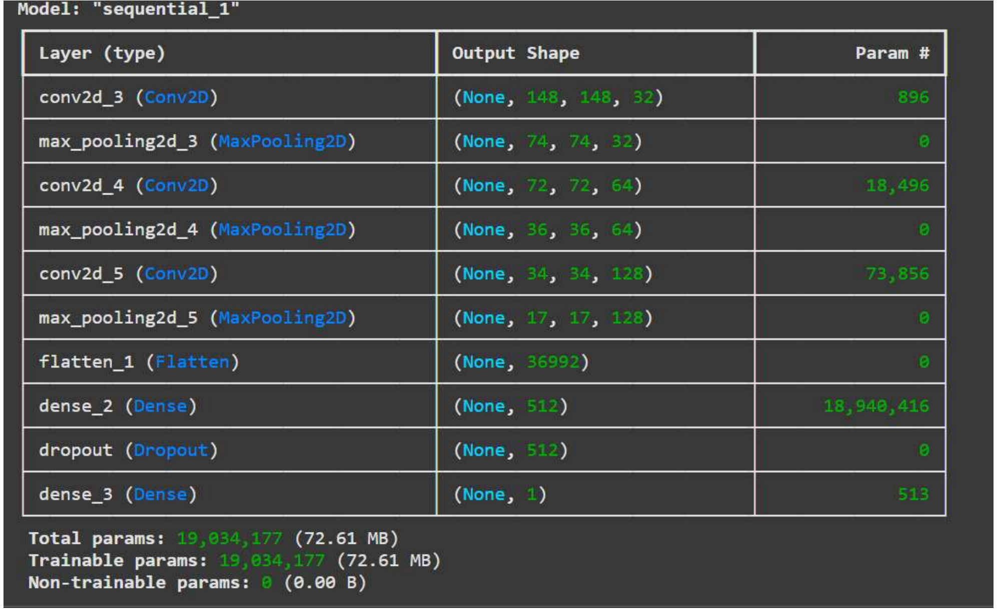
1500 is the larger training sample size.

500 is the size of the validation sample.

500 is the test sample size.

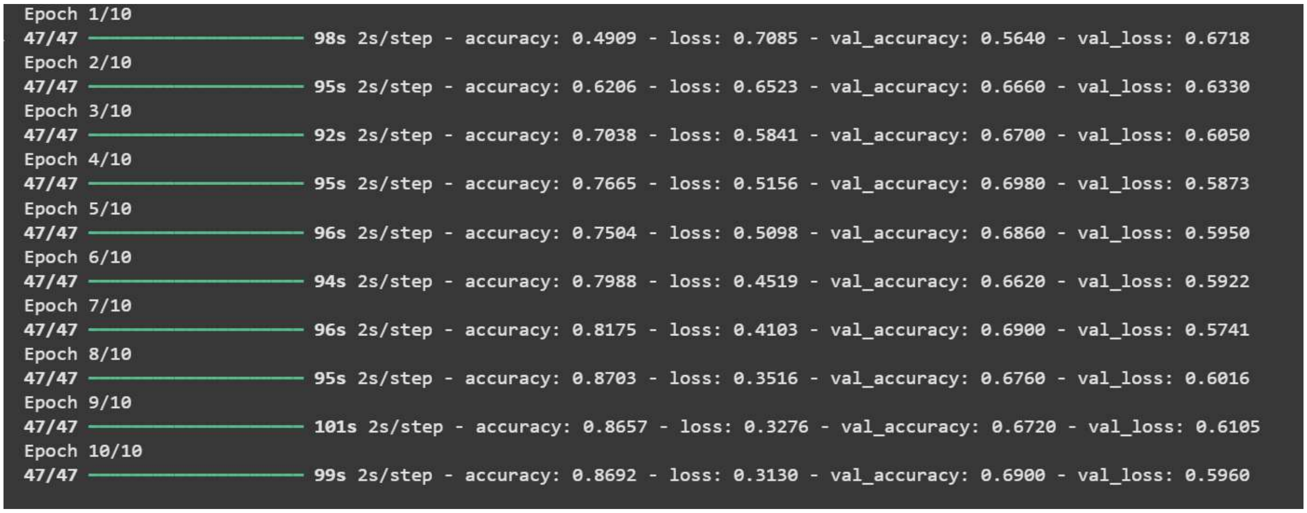
Methods: To lessen overfitting, regularization and data augmentation were used.

Synopsis of the Model:



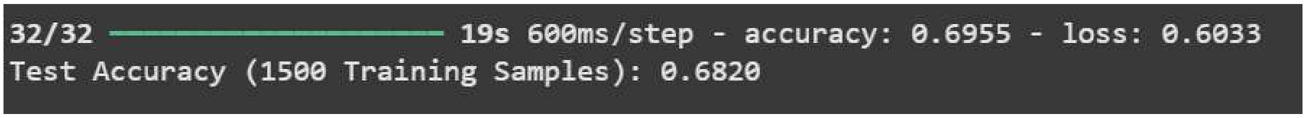
```
Model: "sequential_1"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                         ┃ Output Shape                ┃         Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
│ conv2d_3 (Conv2D)                    │ (None, 148, 148, 32)        │             896 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d_3 (MaxPooling2D)       │ (None, 74, 74, 32)          │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ conv2d_4 (Conv2D)                    │ (None, 72, 72, 64)          │          18,496 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d_4 (MaxPooling2D)       │ (None, 36, 36, 64)          │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ conv2d_5 (Conv2D)                    │ (None, 34, 34, 128)         │          73,856 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ max_pooling2d_5 (MaxPooling2D)       │ (None, 17, 17, 128)         │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ flatten_1 (Flatten)                  │ (None, 36992)               │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_2 (Dense)                      │ (None, 512)                 │      18,940,416 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dropout (Dropout)                    │ (None, 512)                 │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_3 (Dense)                      │ (None, 1)                   │             513 │
└──────────────────────────────────────┴─────────────────────────────┴─────────────────┘
 Total params: 19,034,177 (72.61 MB)
 Trainable params: 19,034,177 (72.61 MB)
 Non-trainable params: 0 (0.00 B)
```

Performance:



```
Epoch 1/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 98s 2s/step - accuracy: 0.4909 - loss: 0.7085 - val_accuracy: 0.5640 - val_loss: 0.6718
Epoch 2/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 95s 2s/step - accuracy: 0.6206 - loss: 0.6523 - val_accuracy: 0.6660 - val_loss: 0.6330
Epoch 3/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 92s 2s/step - accuracy: 0.7038 - loss: 0.5841 - val_accuracy: 0.6700 - val_loss: 0.6050
Epoch 4/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 95s 2s/step - accuracy: 0.7665 - loss: 0.5156 - val_accuracy: 0.6980 - val_loss: 0.5873
Epoch 5/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 96s 2s/step - accuracy: 0.7504 - loss: 0.5098 - val_accuracy: 0.6860 - val_loss: 0.5950
Epoch 6/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 94s 2s/step - accuracy: 0.7988 - loss: 0.4519 - val_accuracy: 0.6620 - val_loss: 0.5922
Epoch 7/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 96s 2s/step - accuracy: 0.8175 - loss: 0.4103 - val_accuracy: 0.6900 - val_loss: 0.5741
Epoch 8/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 95s 2s/step - accuracy: 0.8703 - loss: 0.3516 - val_accuracy: 0.6760 - val_loss: 0.6016
Epoch 9/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 101s 2s/step - accuracy: 0.8657 - loss: 0.3276 - val_accuracy: 0.6720 - val_loss: 0.6105
Epoch 10/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 99s 2s/step - accuracy: 0.8692 - loss: 0.3130 - val_accuracy: 0.6900 - val_loss: 0.5960
```

Result:



```
32/32 ━━━━━━━━━━━━━━━━━━━━ 19s 600ms/step - accuracy: 0.6955 - loss: 0.6033
Test Accuracy (1500 Training Samples): 0.6820
```

Optimal Training Sample Size:

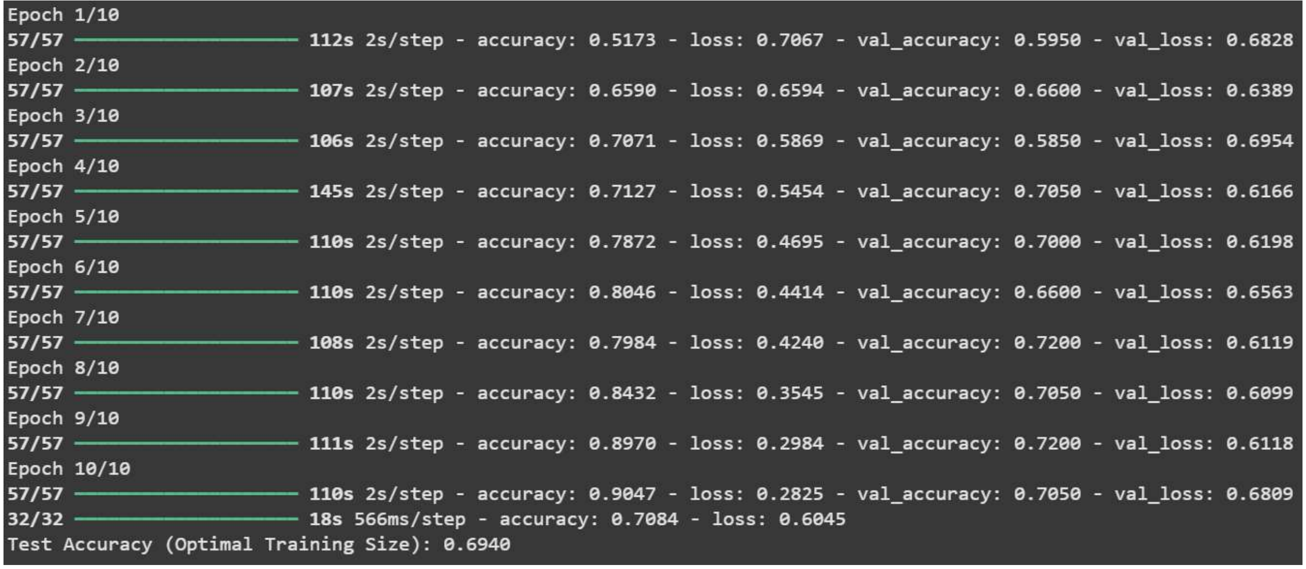
1800 is the larger training sample size.

500 is the size of the validation sample.

500 is the test sample size.

Methods: To lessen overfitting, regularization and data augmentation were used.

Accuracy:



```
Epoch 1/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 112s 2s/step - accuracy: 0.5173 - loss: 0.7067 - val_accuracy: 0.5950 - val_loss: 0.6828
Epoch 2/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 107s 2s/step - accuracy: 0.6590 - loss: 0.6594 - val_accuracy: 0.6600 - val_loss: 0.6389
Epoch 3/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 106s 2s/step - accuracy: 0.7071 - loss: 0.5869 - val_accuracy: 0.5850 - val_loss: 0.6954
Epoch 4/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 145s 2s/step - accuracy: 0.7127 - loss: 0.5454 - val_accuracy: 0.7050 - val_loss: 0.6166
Epoch 5/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 110s 2s/step - accuracy: 0.7872 - loss: 0.4695 - val_accuracy: 0.7000 - val_loss: 0.6198
Epoch 6/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 110s 2s/step - accuracy: 0.8046 - loss: 0.4414 - val_accuracy: 0.6600 - val_loss: 0.6563
Epoch 7/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 108s 2s/step - accuracy: 0.7984 - loss: 0.4240 - val_accuracy: 0.7200 - val_loss: 0.6119
Epoch 8/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 110s 2s/step - accuracy: 0.8432 - loss: 0.3545 - val_accuracy: 0.7050 - val_loss: 0.6099
Epoch 9/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 111s 2s/step - accuracy: 0.8970 - loss: 0.2984 - val_accuracy: 0.7200 - val_loss: 0.6118
Epoch 10/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 110s 2s/step - accuracy: 0.9047 - loss: 0.2825 - val_accuracy: 0.7050 - val_loss: 0.6809
32/32 ━━━━━━━━━━━━━━━━━━━━ 18s 566ms/step - accuracy: 0.7084 - loss: 0.6045
Test Accuracy (Optimal Training Size): 0.6940
```

Using a Pretrained Network:

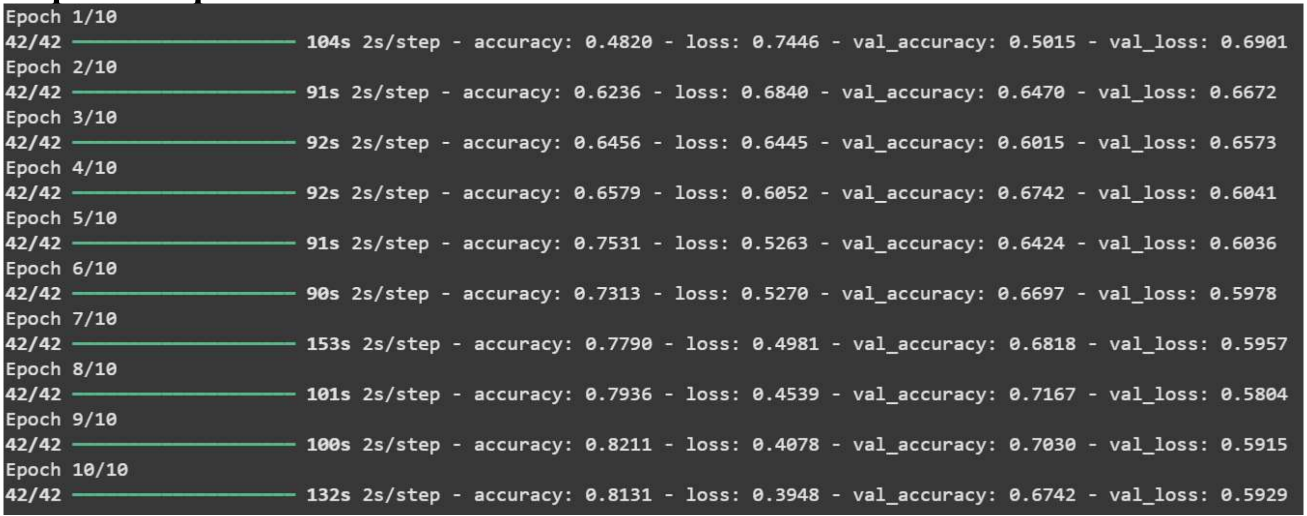
employed the same sample sizes as in Steps 2 and 3 of training from scratch with a pretrained

Performance:

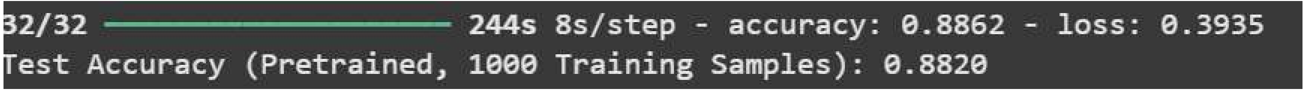
convolutional neural network. used optimization strategies to improve efficiency.

b. Performance with Pretrained Network:

Step 1: Sample Size:1000

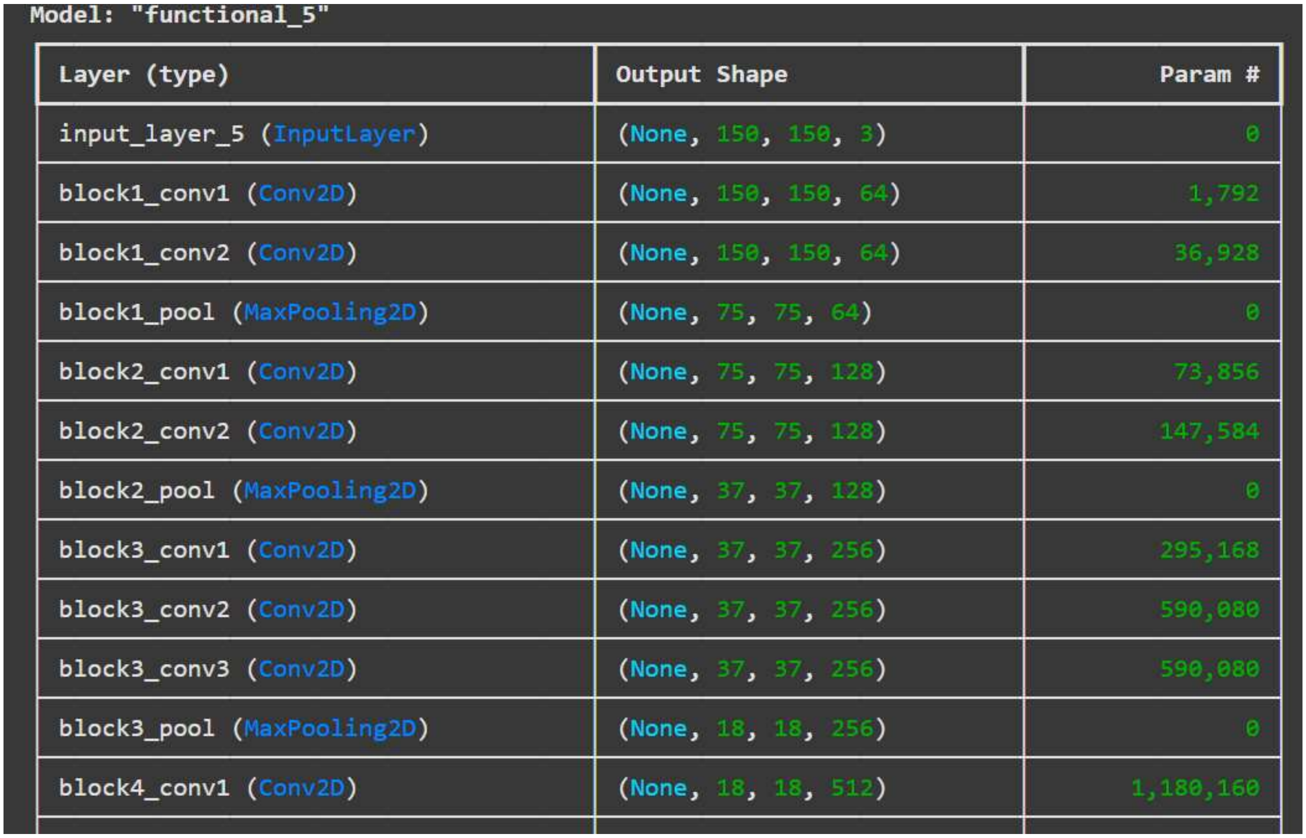


```
Epoch 1/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 104s 2s/step - accuracy: 0.4820 - loss: 0.7446 - val_accuracy: 0.5015 - val_loss: 0.6901
Epoch 2/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 91s 2s/step - accuracy: 0.6236 - loss: 0.6840 - val_accuracy: 0.6470 - val_loss: 0.6672
Epoch 3/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 92s 2s/step - accuracy: 0.6456 - loss: 0.6445 - val_accuracy: 0.6015 - val_loss: 0.6573
Epoch 4/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 92s 2s/step - accuracy: 0.6579 - loss: 0.6052 - val_accuracy: 0.6742 - val_loss: 0.6041
Epoch 5/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 91s 2s/step - accuracy: 0.7531 - loss: 0.5263 - val_accuracy: 0.6424 - val_loss: 0.6036
Epoch 6/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 90s 2s/step - accuracy: 0.7313 - loss: 0.5270 - val_accuracy: 0.6697 - val_loss: 0.5978
Epoch 7/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 153s 2s/step - accuracy: 0.7790 - loss: 0.4981 - val_accuracy: 0.6818 - val_loss: 0.5957
Epoch 8/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 101s 2s/step - accuracy: 0.7936 - loss: 0.4539 - val_accuracy: 0.7167 - val_loss: 0.5804
Epoch 9/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 100s 2s/step - accuracy: 0.8211 - loss: 0.4078 - val_accuracy: 0.7030 - val_loss: 0.5915
Epoch 10/10
42/42 ━━━━━━━━━━━━━━━━━━━━ 132s 2s/step - accuracy: 0.8131 - loss: 0.3948 - val_accuracy: 0.6742 - val_loss: 0.5929
```



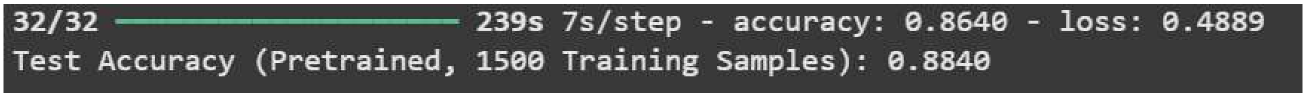
```
32/32 ━━━━━━━━━━━━━━━━━━━━ 244s 8s/step - accuracy: 0.8862 - loss: 0.3935
Test Accuracy (Pretrained, 1000 Training Samples): 0.8820
```

Step 2: Sample Size: 1500



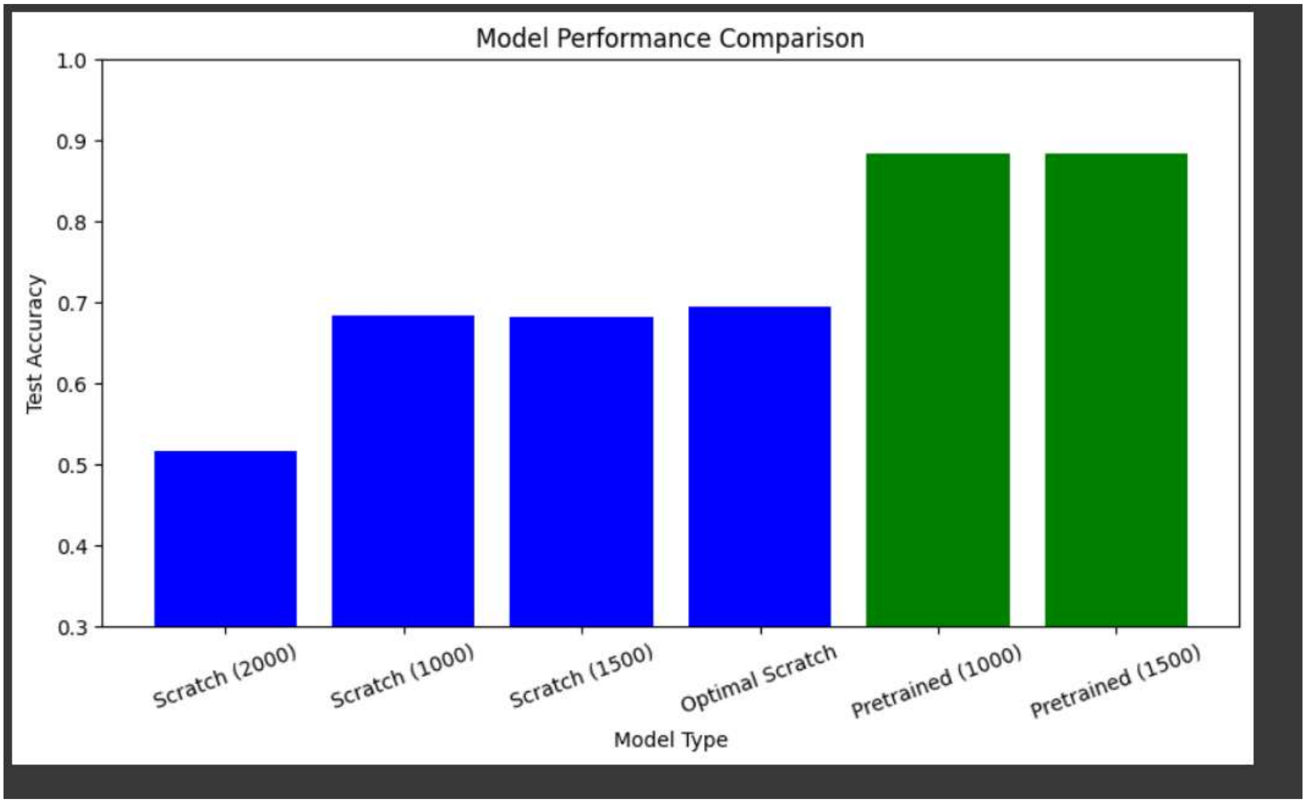
```
Model: "functional_5"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ input_layer_5 (InputLayer)      │ (None, 150, 150, 3)    │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block1_conv1 (Conv2D)           │ (None, 150, 150, 64)   │         1,792 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block1_conv2 (Conv2D)           │ (None, 150, 150, 64)   │        36,928 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block1_pool (MaxPooling2D)      │ (None, 75, 75, 64)     │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block2_conv1 (Conv2D)           │ (None, 75, 75, 128)    │        73,856 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block2_conv2 (Conv2D)           │ (None, 75, 75, 128)    │       147,584 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block2_pool (MaxPooling2D)      │ (None, 37, 37, 128)    │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block3_conv1 (Conv2D)           │ (None, 37, 37, 256)    │       295,168 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block3_conv2 (Conv2D)           │ (None, 37, 37, 256)    │       590,080 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block3_conv3 (Conv2D)           │ (None, 37, 37, 256)    │       590,080 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block3_pool (MaxPooling2D)      │ (None, 18, 18, 256)    │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ block4_conv1 (Conv2D)           │ (None, 18, 18, 512)    │     1,180,160 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
```

Accuracy:



```
32/32 ━━━━━━━━━━━━━━━━━━━━ 239s 7s/step - accuracy: 0.8640 - loss: 0.4889
Test Accuracy (Pretrained, 1500 Training Samples): 0.8840
```

Final Graph: -



Findings:

Training from Scratch:

Accuracy increases as the ideal training sample size was reached. Regularization and data augmentation were essential for reducing overfitting.

Using a Pretrained Network:

In general, pretrained networks performed better than scratch-trained networks. Although pretrained networks were more resilient to lower datasets, the effect of training sample size on performance was still noticeable. Utilizing knowledge from pretrained models was successfully accomplished through transfer learning. The connection between network choice and training sample size is the following findings clearly show how training sample size and network selection are related:

Training from Scratch:

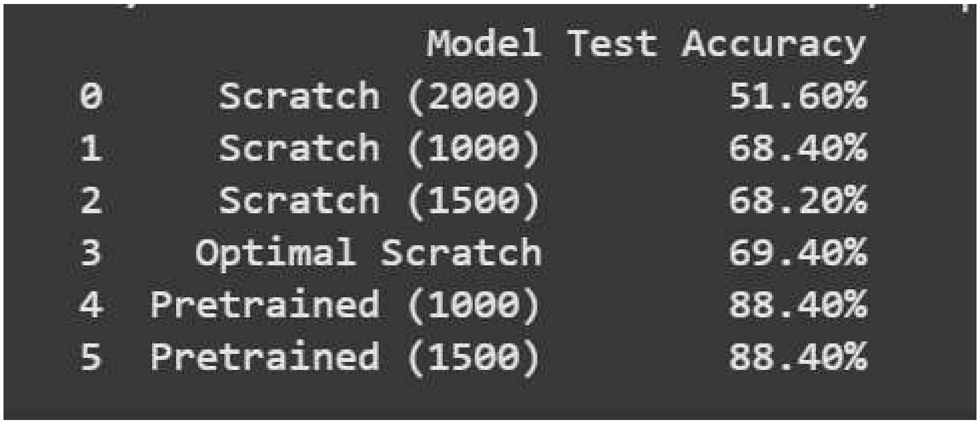
Achieving the best outcomes requires optimal training sample sizes.

Using a Pretrained Network:

Higher resistance to smaller training sample sizes was shown by pretrained networks. When compared to training from scratch, the model performed better thanks to transfer learning, which enabled it to use knowledge from pretrained architecture.

Summary:

In conclusion, the available training sample size affects the network selection, whether training from scratch or utilizing a pretrained convnet. Both strategies gain from larger datasets, but when training data is limited, pretrained networks provide a reliable solution. Achieving optimal performance in picture classification tasks requires an understanding of the link between sample size and network choice, as the table below illustrates:



```
                Model Test Accuracy
0     Scratch (2000)         51.60%
1     Scratch (1000)         68.40%
2     Scratch (1500)         68.20%
3    Optimal Scratch         69.40%
4  Pretrained (1000)         88.40%
5  Pretrained (1500)         88.40%
```