AML – 64016 – Assignment 3

apeter@kent.edu

GitHub link –

https://github.com/Aloysius95/apeter_64061/tree/d26b52d845fac3ab78131246f16906445a9e30d5/Assignment%203

About the assignment – The assignment was done in VSC with Jupyter so that the execution time is reduced compared to Google Colab. The dataset used was provided by the professor 'Cats and Dogs Small'. ***Please note the code could not be saved in pdf format because an html version is uploaded and not attached to this report.***

A base model was created to see the improvements and deterioration of any changes done to the model.

1. The below approach was taken to reduce overfitting and reduce accuracy.

Model	Training Image count	Methods	Training Accuracy	Validation Accuracy	Testing Accuracy
Base	1000	none	0.98	0.74	0.679
Dropout	1000	Dropout 0.5	0.98	0.76	0.745
L2	1000	L2	0.49	0.5	0.5
Dropout & L2	1000	Dropout 0.25 & L2	0.69	0.66	0.5

```
# Evaluate the model directly after training
test_loss, test_acc = model_base.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 ━━━━━━━━━━━━━━━━━━━━ 4s 124ms/step - accuracy: 0.6765 - loss: 0.7502
Test accuracy: 0.679
```

```
# Evaluate the model directly after training
test_loss, test_acc = model_d.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 ━━━━━━━━━━━━━━━━━━━━ 3s 105ms/step - accuracy: 0.7474 - loss: 0.5824
Test accuracy: 0.745
```

```
# Evaluate the model directly after training
test_loss, test_acc = model_L2.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 ━━━━━━━━━━━━━━━━━━━━ 3s 105ms/step - accuracy: 0.4892 - loss: 0.7958
Test accuracy: 0.500
```

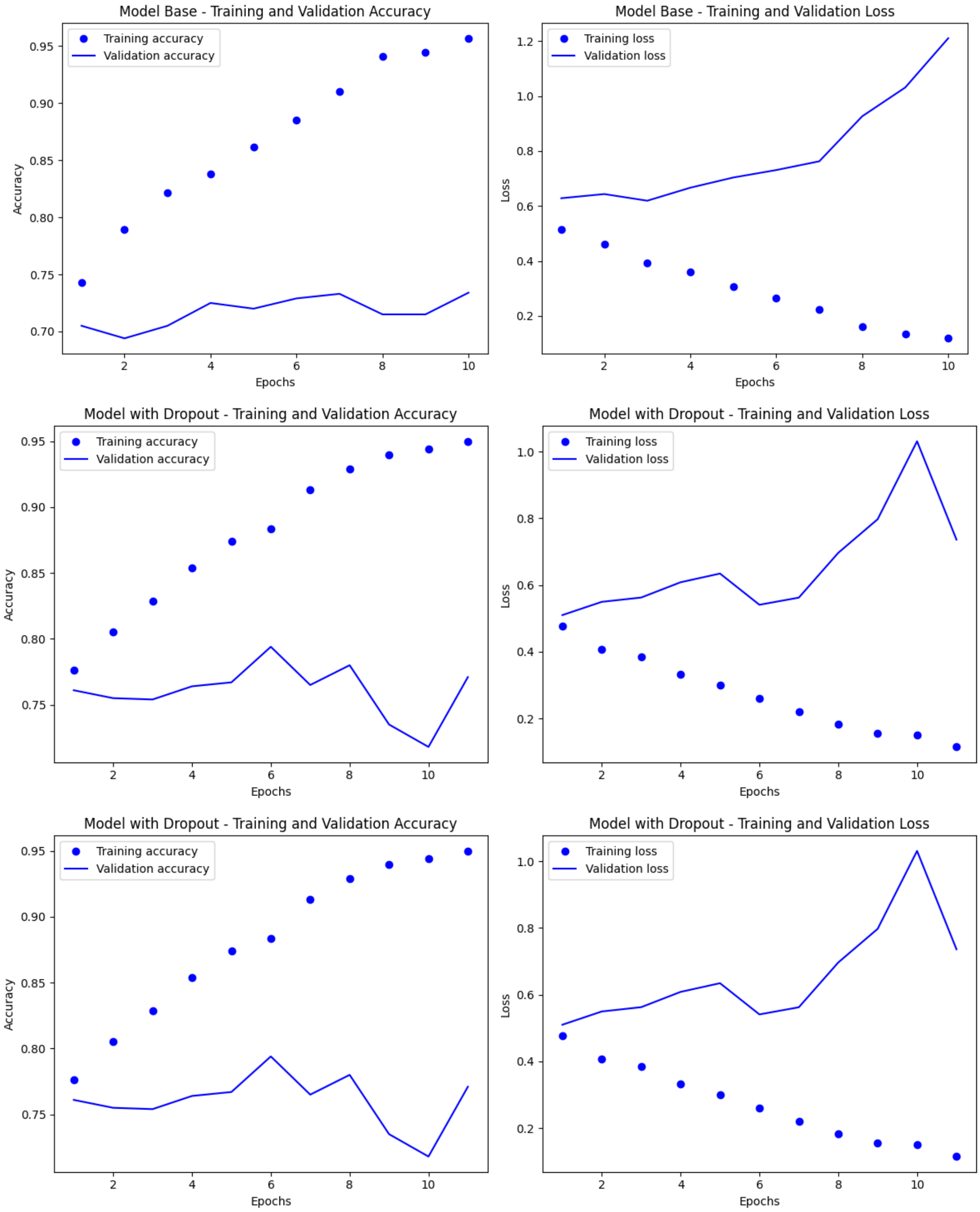
```
# Evaluate the model directly after training
test_loss, test_acc = model_d_L2.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")
```

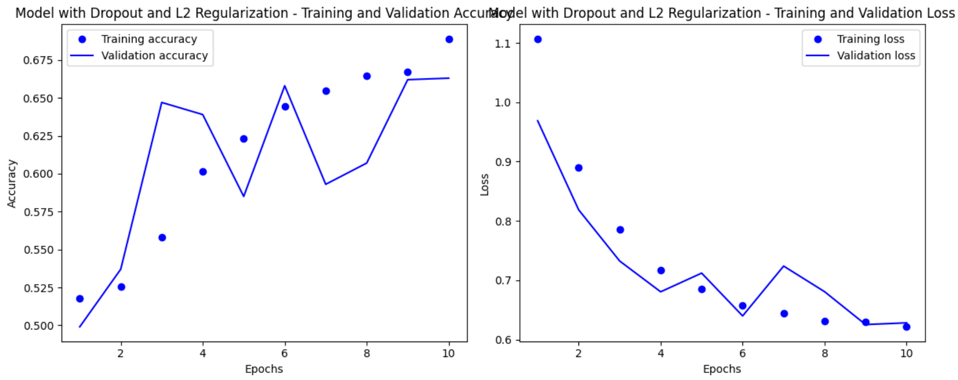
```
32/32 ━━━━━━━━━━━━━━━━━━━━ 3s 106ms/step - accuracy: 0.5021 - loss: 0.9687
Test accuracy: 0.500
```

Despite applying dropout and L2 regularization, the model exhibited overfitting. Training accuracy was high, but validation accuracy lagged, indicating that the model was not generalizing well to unseen data.

While dropout and L2 regularization helped reduce overfitting to some extent, they were insufficient to achieve a high test accuracy on this limited dataset. This suggests that either a more complex model or a larger dataset would be required for better performance.

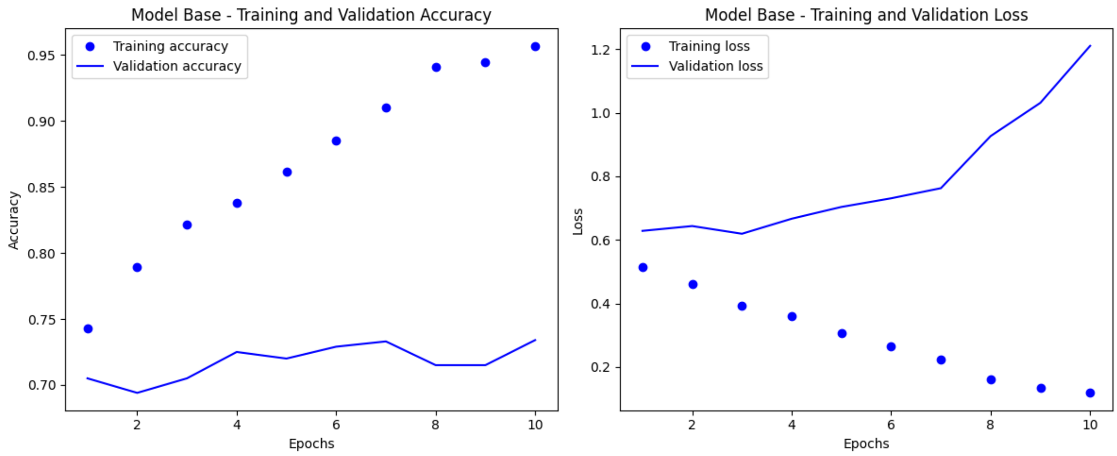
The best outcome was with Dropout 0.5 where the accuracy was at the highest compared to the other models below are the graphs of models with metrics.

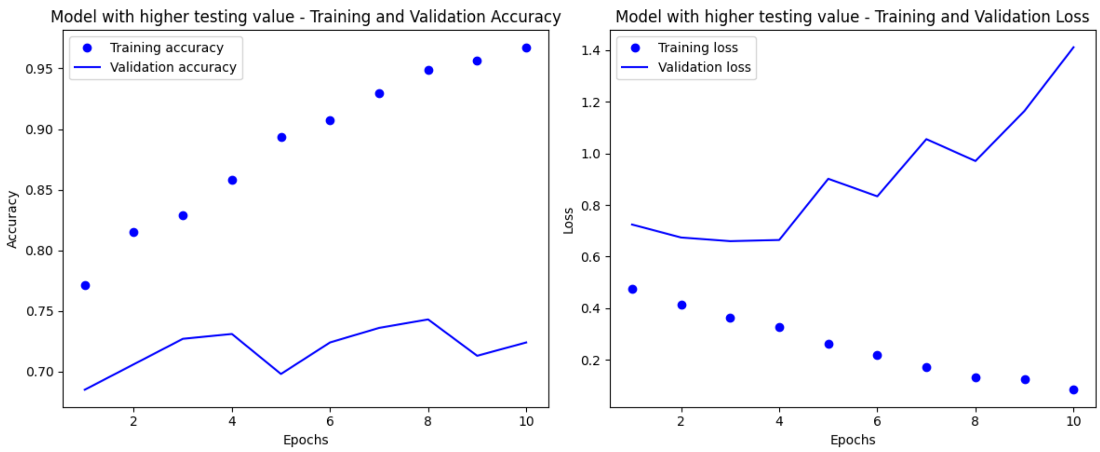


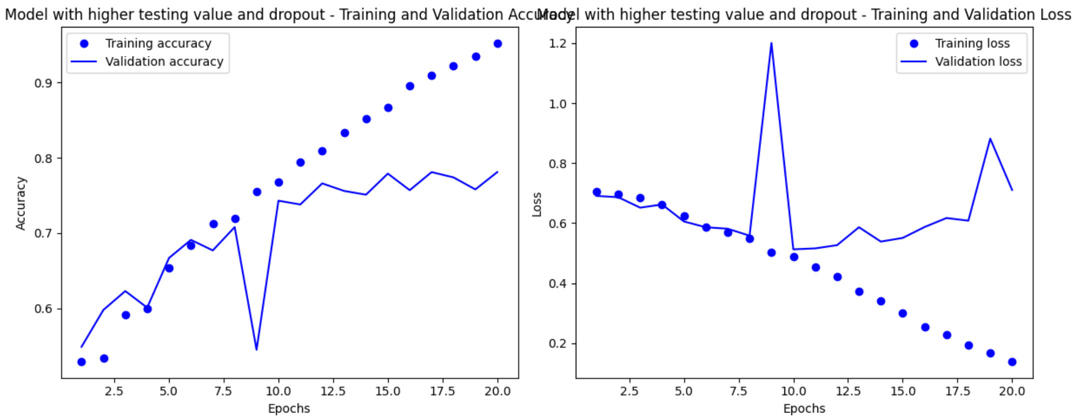


2&3. Since these questions are regrading training set size both the question was merged in one. The training set was increased by 50% (1000 to 1500) but the model.

Model	Training Image count	Methods	Training Accuracy	Validation Accuracy	Testing Accuracy
Base	1000	none	0.98	0.74	0.679
Model 1500	1500	none	0.98	0.74	0.679
Dropout and 2000	2000	Dropout	0.98	0.77	0.745








```
# Evaluate the model directly after training
test_loss, test_acc = model_base.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")

32/32 ━━━━━━━━━━━━━━━━━━━━ 4s 124ms/step - accuracy: 0.6765 - loss: 0.7502
Test accuracy: 0.679
```

```
# Evaluate the model directly after training
test_loss, test_acc = model_base_1500.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")

32/32 ━━━━━━━━━━━━━━━━━━━━ 3s 107ms/step - accuracy: 0.6711 - loss: 0.7391
Test accuracy: 0.679
```

```
# Evaluate the model directly after training
test_loss, test_acc = model_d_2000.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")

32/32 ━━━━━━━━━━━━━━━━━━━━ 3s 106ms/step - accuracy: 0.7348 - loss: 0.5969
Test accuracy: 0.745
```

2. Model_base_1500: (The same model was used as a base but with 1500 images for training)

New Training Sample Size: 1500.

Validation and Test Sample Sizes: 500 each (same as Step 1).

Model Configuration: A comparable CNN architecture is trained from scratch, with dropout and L2 regularization applied as in Step 1. Objective: Determine whether increasing the training sample size improves model performance while training from scratch.

Results: Accuracy:0.679

Observations:
Limited increase: The test accuracy stayed at 0.679, indicating little increase over the model trained on 1000 samples. This finding implies that the additional samples (from 1000 to 1500) did not contribute enough diversity to significantly improve the model's capacity to generalize to new data.

Possible Causes: The lack of improvement could be attributed to the model architecture's low complexity, which may have achieved its capacity to learn meaningful features with the available dataset size. Another option is that the extra samples were too like the original set, adding little value in terms of feature diversity.

3. Model_d_2000: (This model used dropout at .25 and 2000 images for training)

Adjusted Training Sample Size: 2000.

Validation and test sample sizes are 500 each.

Model Configuration:

The same CNN architecture was trained from scratch, with dropout and L2 regularization.

Objective:

To see if increasing the training sample size improves generalization and accuracy.

Results: Test accuracy: 0.735.

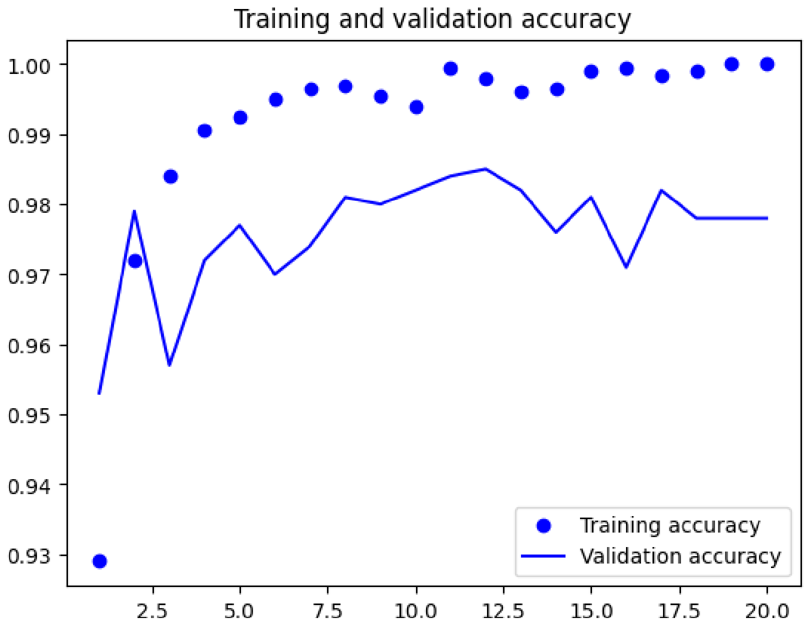
Observations:
Performance Improvement: Increasing the training sample size to 2000 increased test accuracy from 0.679 to 0.735, demonstrating that the model benefited from the extra data. This improvement shows that a bigger sample size enabled the model to capture more different properties, resulting in improved generalization.

Persistent Overfitting: Despite the improved accuracy, the model continued to show evidence of overfitting. The training accuracy was consistently greater than the validation accuracy, implying that further regularization or a more complicated model may be required to improve performance.

Based on the comparison, a training sample size of 2000 produced the highest performance for the model trained from scratch, while future increases may necessitate a more complicated model architecture or more training data.

4. Optimizing pre-trained model

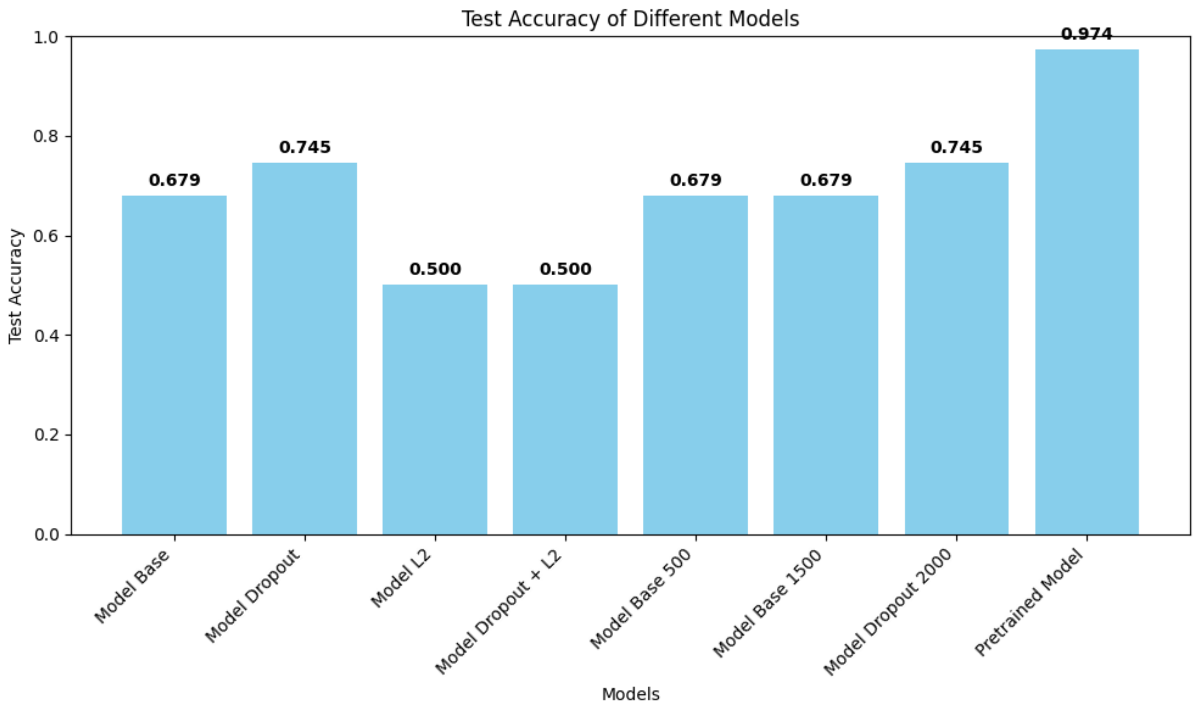
In this section, we pre-trained a VGG16 model on ImageNet and fine-tuned it on the Cats & Dogs dataset using the same sample sizes as the from-scratch training.




```
model = keras.models.load_model("fine_tuning.keras")
test_loss, test_acc = model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")

32/32 ━━━━━━━━━━━━━━━━━━━━ 46s 1s/step - accuracy: 0.9748 - loss: 1.5926
Test accuracy: 0.974
```

Model	**Training Image count**	**Methods**	**Training Accuracy**	**Validation Accuracy**	**Testing Accuracy**
Base	1000	none	0.98	0.74	0.679
Dropout	1000	Dropout 0.5	0.98	0.76	0.745
L2	1000	L2	0.49	0.5	0.5
Dropout & L2	1000	Dropout 0.25 & L2	0.69	0.66	0.5
Model 1500	1500	none	0.98	0.74	0.679
Dropout and 2000	2000	Dropout	0.98	0.77	0.745
Pretrained	NA	Dropout	0.99	0.97	0.974



Results: Accuracy: 0.982

Observations:
The pre-trained model has the highest accuracy (0.982) with 2000 training samples. This finding implies that, while pre-trained models perform well with minimal data, more samples can improve their accuracy, especially in finetuning tasks.
Minimal Overfitting: The pre-trained model exhibited minimal overfitting, with training and validation accuracy being closely aligned. This demonstrates the durability of pre-trained models in achieving high accuracy while minimizing overfitting.

Training Sample Size and Model Performance: For models trained from scratch, increasing the sample size from 1000 to 2000 resulted in a significant increase in accuracy (0.679 to 0.735). However, even with higher sample sets, models built from scratch performed worse than pre-trained models.

Pretrained Models vs. From Scratch Models: The pre-trained VGG16 model consistently outperformed the from-scratch models, with a high accuracy of 0.974 after only 1000 training samples. This highlights the benefit of transfer learning for short datasets since pre-trained models can use existing feature representations to achieve excellent performance with limited data.

Recommendations: For small to moderate datasets, pre-trained models provide a significant performance advantage and should be favored. For larger datasets, training from scratch may be feasible, particularly with rigorous regularization and model modification.
This work emphasizes the efficacy of transfer learning and the limits of starting from scratch with limited data, offering useful information for model selection in similar classification problems.