Assignment 3 – AML

Student Name: Aksa Taniya

Instructor: CJ Wu

Topic: Cats Vs Dogs Classification using Deep Learning

For this task, I have classified the following into following cases as per the instructions provided.

I categorized the use cases into the following,

1. Basic Convolution network from the scratch
2. Convolution network with optimization techniques
3. Using pre-trained network

And then sub categorized depending on the training samples.

I used **ResNet50** architecture as my pretrained network and a brief detail provided below

ResNet50 is a convolutional neural network architecture that was proposed by Microsoft Research in 2015. It is one of the variants of the ResNet (Residual Network) family, which introduced the concept of residual learning to improve the training of very deep neural networks.

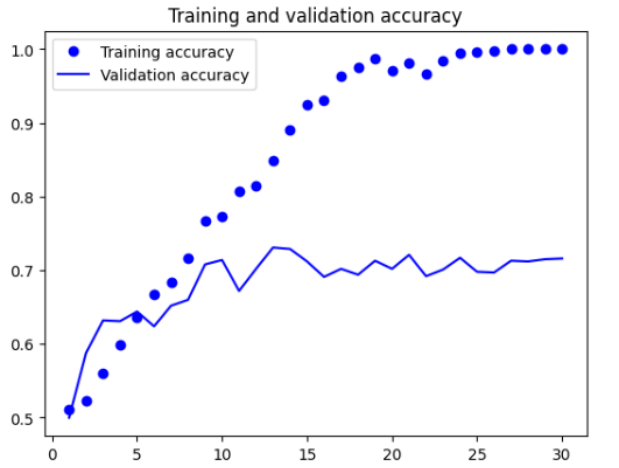
ResNet50 consists of 50 layers, including a convolutional layer, followed by four stages, each containing multiple residual blocks. Each residual block consists of two convolutional layers and a shortcut connection that bypasses the convolutional layers. The shortcut connection allows for the gradient to flow directly through the network without being affected by the convolutional layers, which can help to mitigate the vanishing gradient problem and improve the training of very deep neural networks.

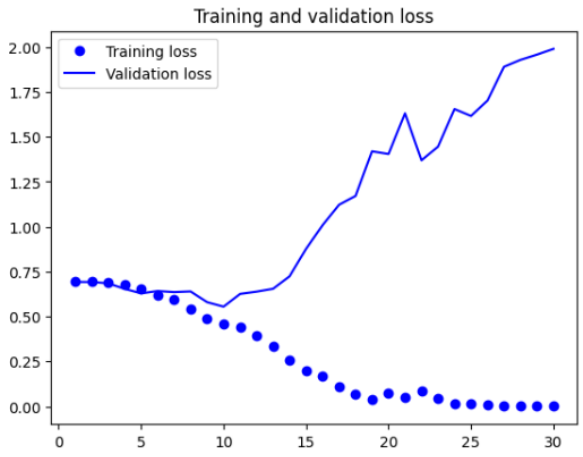
ResNet50 also includes a global average pooling layer and a fully connected layer at the end of the network, which outputs the class scores for the input image. The architecture has been pre-trained on a large dataset such as ImageNet and can be fine-tuned for specific tasks such as image classification, object detection, and segmentation

And the results are displayed below.

Case: 1 - Convnet from Scratch with 1000 sample data and 500 validation and testing data

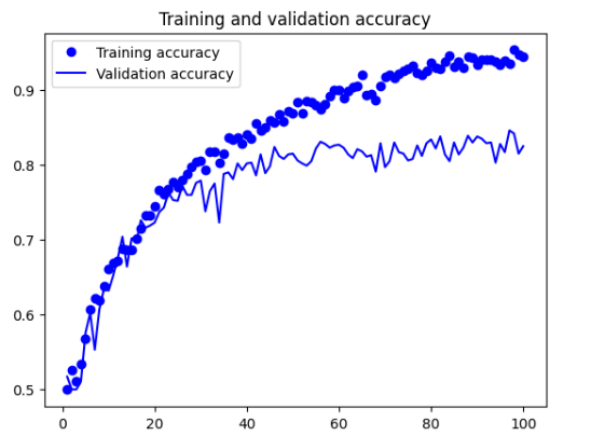
```
Epoch 30/30
63/63 [==============================] - 4s 60ms/step - loss: 2.6084e-04 - accu
racy: 1.0000 - val_loss: 1.9900 - val_accuracy: 0.7160
```

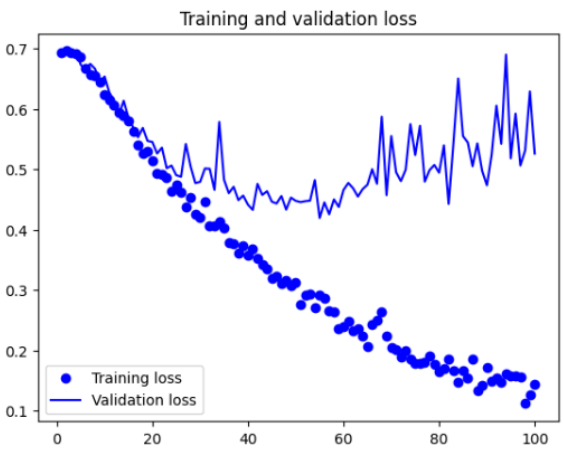




Case: 2 - Convnet from scratch with 1000 sample data and 500 validation and testing data also with Dropout and Data Augmentation – (optimization Techniques)

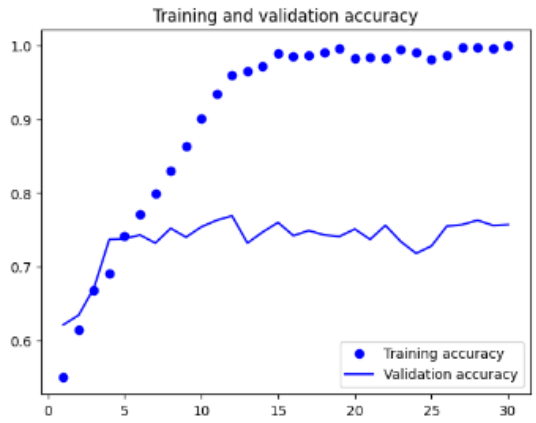
```
Epoch 100/100
63/63 [==============================] - 4s 62ms/step - loss: 0.1434 - accurac
y: 0.9445 - val_loss: 0.5258 - val_accuracy: 0.8250
```

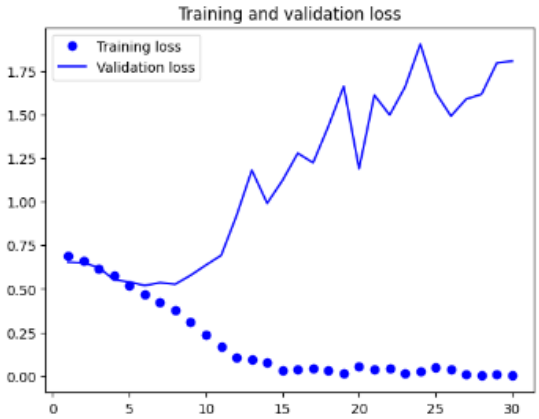




Case: 3 - Convnet from Scratch with 2000 training samples and 500 validation and testing data.

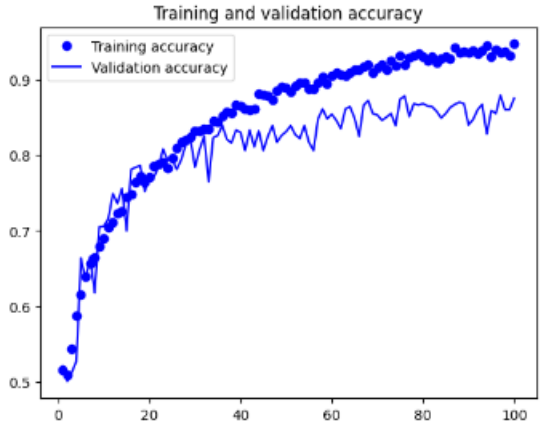
```
Epoch 30/30
125/125 [==============================] - 7s 53ms/step - loss: 0.0014 - accuracy: 0.9998 - val_loss:
1.8062 - val_accuracy: 0.7570
```

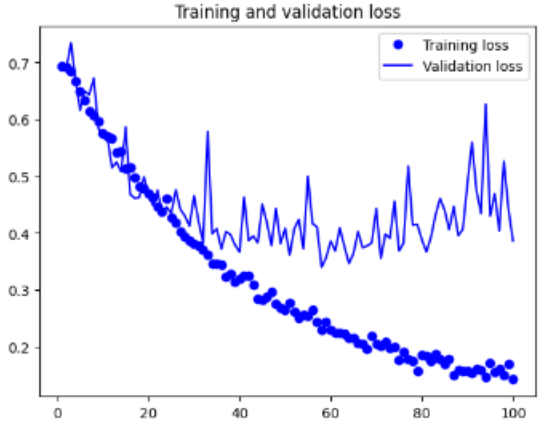




Case: 4 - Convnet with Dropout and Data Augmentation with 2000 training samples and 500 validation and testing data – (optimization Techniques)

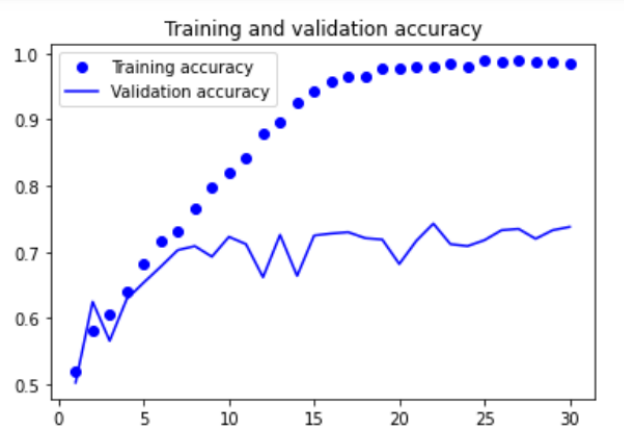
```
Epoch 100/100
125/125 [==============================] - 7s 56ms/step - loss: 0.1421 - accuracy: 0.9467 - val_los
s: 0.3861 - val_accuracy: 0.8750
```

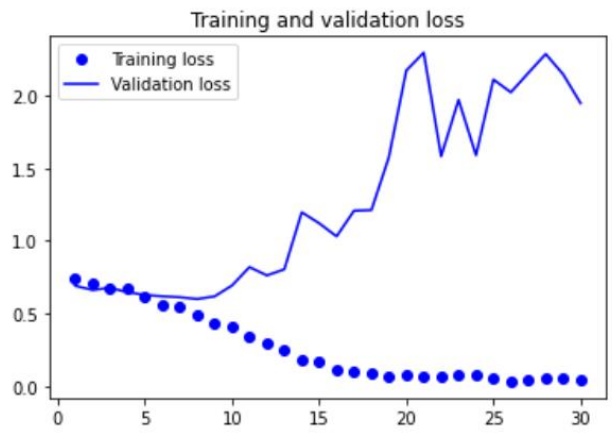




Case: 5- Convnet from Scratch with 5000 training samples and 500 validation and testing data

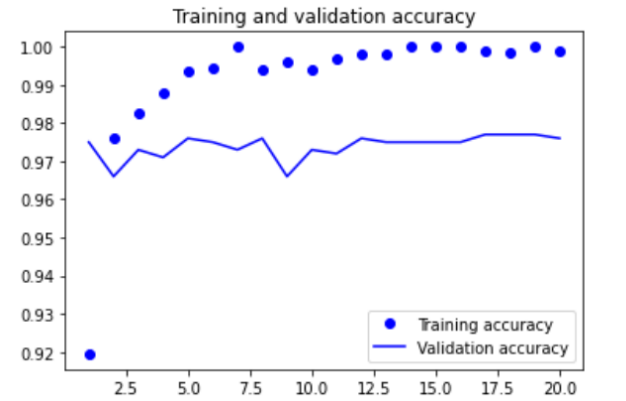
```
2.1492 - val_accuracy: 0.7350
Epoch 30/30
63/63 [==============================] - 5s 74ms/step - loss: 0.0420 - accuracy: 0.9840 - val_loss:
1.9500 - val_accuracy: 0.7380
```

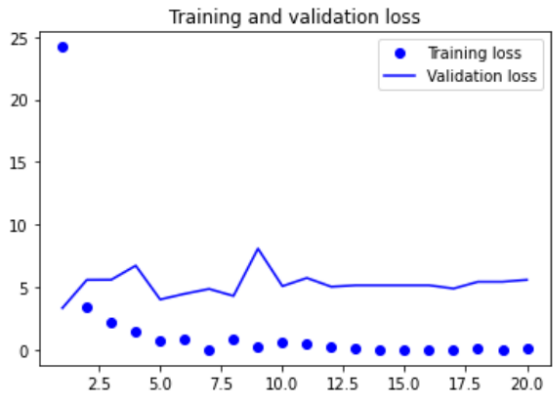




Case: 6 - Convnet with Dropout and Data Augmentation with 5000 training samples and 500 validation and testing data – (optimization Techniques)

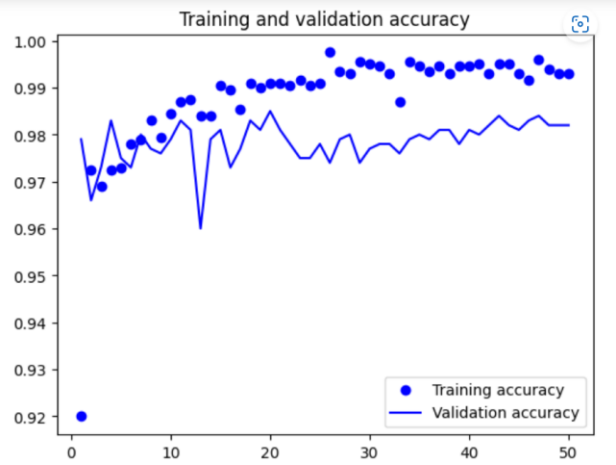
```
Epoch 100/100
63/63 [==============================] - 6s 99ms/step - loss: 0.2075 - accuracy: 0.9310 - val_loss:
0.6068 - val_accuracy: 0.8410
```

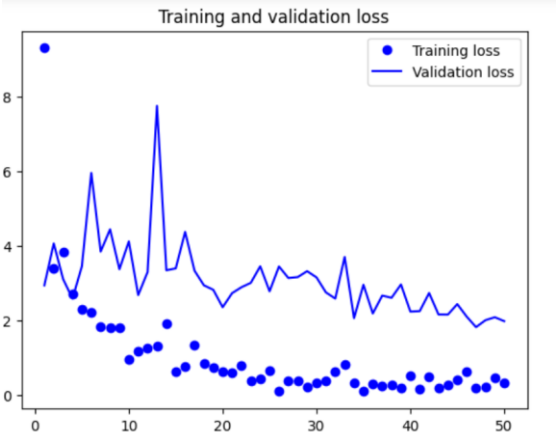




Case: 7 - Using pre-trained ResNet50 architecture with 1000 sample data and 500 validation and testing data

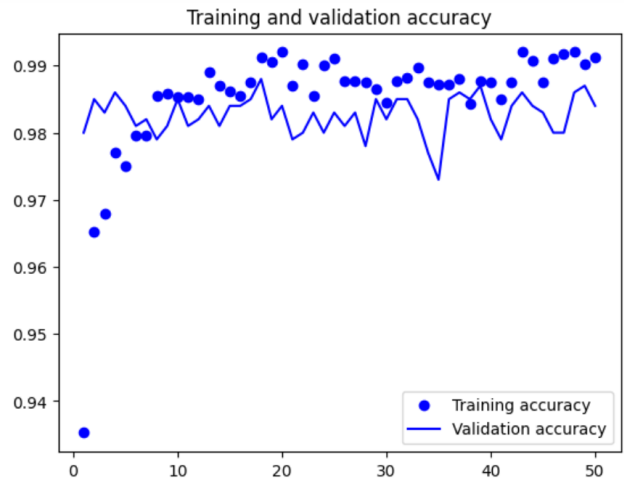
```
Epoch 50/50
63/63 [==============================] - 8s 123ms/step - loss: 0.3459 - accuracy: 0.9930 - val_loss:
1.9872 - val_accuracy: 0.9820
```

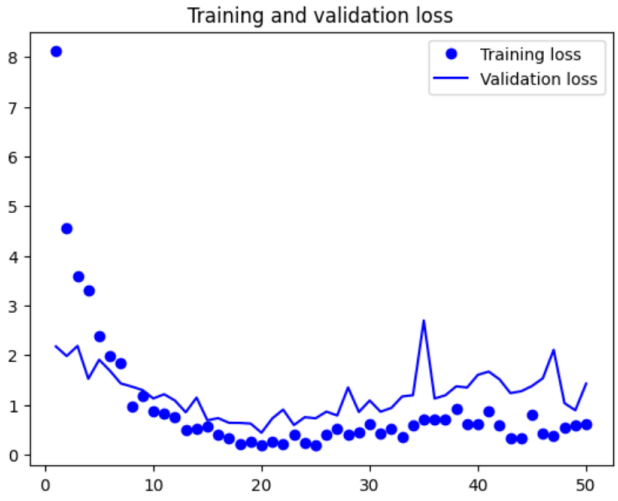




Case: 8 - Using pre-trained ResNet50 architecture with 2000 sample data and 500 validation and testing data

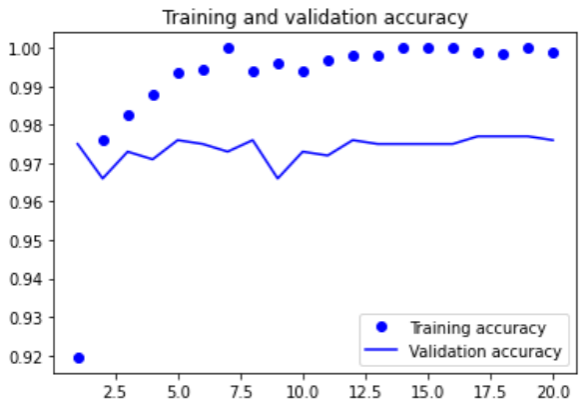
```
Epoch 50/50
125/125 [==============================] - 13s 103ms/step - loss: 0.6222 - accuracy: 0.9912 - val_lo
ss: 1.4262 - val_accuracy: 0.9840
```

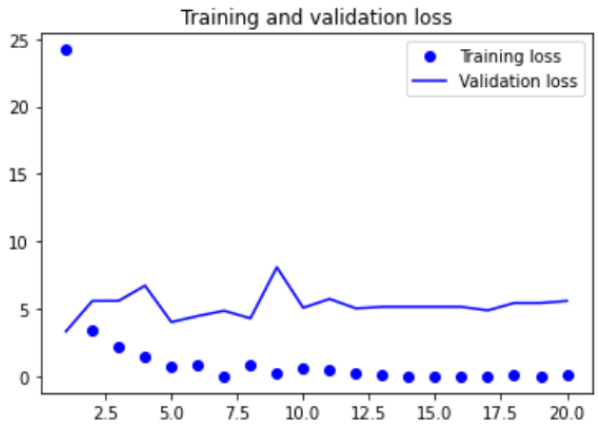




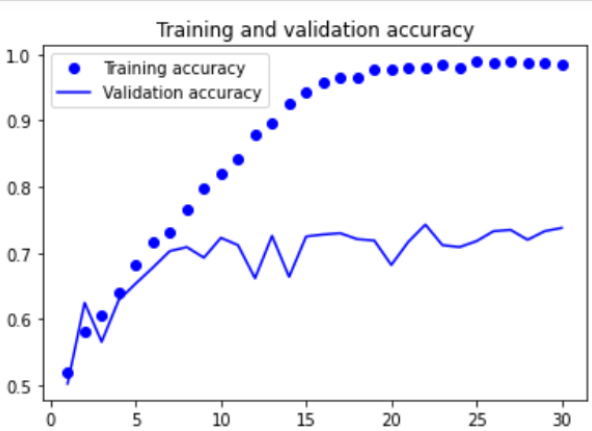
Case: 9 - Using pre-trained ResNet50 architecture with 5000 sample data and 500 validation and testing data

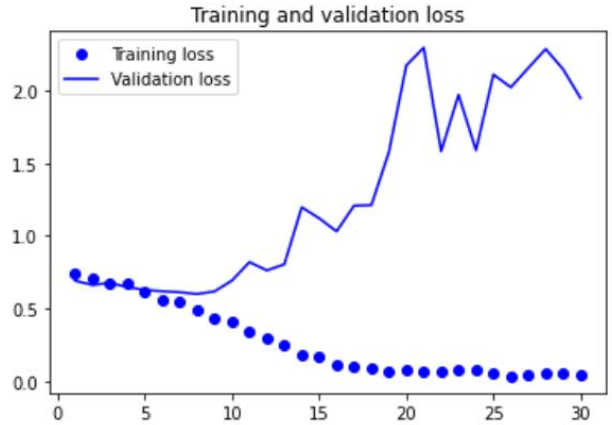
```
s: 2.3566 - val_accuracy: 0.9840
Epoch 50/50
63/63 [==============================] - 13s 210ms/step - loss: 0.4243 - accuracy: 0.9905 - val_los
s: 2.5200 - val_accuracy: 0.9800
```





Case: 10 - Using pre-trained ResNet50 architecture with 5000 sample data and 500 validation and testing data along with optimization techniques.





Summary of Results:

Cases	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Test accuracy
Case - 1	1.00	0.716	2.608	1.990	0.772
Case - 2	0.9445	0.9445	0.1434	0.5258	0.813
Case - 3	0.9998	0.7570	0.0014	0.0014	0.721

Case - 4	0.9467	0.8750	0.1421	0.3861	0.838
Case - 5	0.9848	0.7380	0.0420	1.9500	0.782
Case -6	0.9310	0.8410	0.2075	0.6068	0.834
Case -7	0.9930	0.9820	0.3459	1.9872	0.978
Case-8	0.991	0.9840	0.622	1.426	0.980
Case -9	0.9990	0.9760	0.1155	5.5881	0.9765
Case -10	0.9905	0.9800	0.4243	2.5200	0.9790

Conclusion:

Using a basic convolution, the accuracy of the model is normal, and it touches around 70 percent, As we performed the optimizations techniques and increases the sample size, it varied a lot and increased, thus proving the fact that these techniques really worked in practical. In later stage I used pretrained Resnet50 architecture which gives a pretty decent result and great improvement in the performance.

Overall, the highest performance was achieved is 98 percent, Thus the model can predict the classification task with 98 percent accuracy.