Advanced Machine Learning Assignment 2 (Convolution)

Uploading Kaggle API File and Downloading Dogs vs Cats dataset from Kaggle

In [1]:
```
from google.colab import files
files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving kaggle.json to kaggle.json
```

Out[1]:
```
{'kaggle.json': b'{"username":"ruthvick","key":"78858cf417af4f8f837ed7131aa8287
d"}'}
```

In [2]:
```
!mkdir ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
```

In [3]:
```
!kaggle competitions download -c dogs-vs-cats
!unzip -qq dogs-vs-cats.zip
!unzip -qq train.zip
```

```
Downloading dogs-vs-cats.zip to /content
 98% 795M/812M [00:03<00:00, 219MB/s]
100% 812M/812M [00:03<00:00, 251MB/s]
```

Q1. Consider the Cats & Dogs example. Start initially with a training sample of 1000, a validation sample of 500, and a test sample of 500 (like in the text). Use any technique to reduce overfitting and improve performance in developing a network that you train from scratch. What performance did you achieve?

Creating and Copying dataset to test, train and validation directory

In [4]:
```
import os, shutil, pathlib
o_dir = pathlib.Path("train")
n_dir = pathlib.Path("cats_vs_dogs_small")

def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = n_dir / subset_name / category
        os.makedirs(dir, exist_ok=True)

        fnames = [f"{category}.{i}.jpg" for i in range(start_index, end_index)]
        for fname in fnames:
            src = o_dir / fname
            dst = dir / fname
            shutil.copyfile(src, dst)
```

```
make_subset("train", start_index=500, end_index=1500)
make_subset("validation", start_index=1500, end_index=2000)
make_subset("test", start_index=2000, end_index=2500)
```

Building a basic model to classify dogs and cats using convolutional neural networks

In [5]:

```
from tensorflow.keras.utils import image_dataset_from_directory

train_data = image_dataset_from_directory(n_dir / "train",image_size=(180, 180),bat

valid_data = image_dataset_from_directory(n_dir / "validation",image_size=(180, 180

test_data= image_dataset_from_directory(n_dir / "test",image_size=(180, 180),batch_
```

```
Found 2000 files belonging to 2 classes.
Found 1000 files belonging to 2 classes.
Found 1000 files belonging to 2 classes.
```

Create an instance of the dataset using a NumPy array that has 1000 random samples with a vector size of 16

In [6]:

```
import numpy as np
import tensorflow as tf

ran_num = np.random.normal(size=(1000, 16))
dataset = tf.data.Dataset.from_tensor_slices(ran_num)
for i, element in enumerate(dataset):
    print(element.shape)
    if i >= 2:
       break

batched_dataset = dataset.batch(32)
for i, element in enumerate(batched_dataset):
    print(element.shape)
    if i >= 2:
       break

reshaped_dataset = dataset.map(lambda x: tf.reshape(x, (4, 4)))
for i, element in enumerate(reshaped_dataset):
    print(element.shape)
    if i >= 2:
       break
```

```
(16,)
(16,)
(16,)
(32, 16)
(32, 16)
(32, 16)
(4, 4)
(4, 4)
(4, 4)
```

Displaying the shapes of the data and labels yielded by the Dataset

In [7]:

```
for dataset_batch, label_batch in train_data:
    print("data batch shape:", dataset_batch.shape)
    print("labels batch shape:", label_batch.shape)
    break
```

```
data batch shape: (32, 180, 180, 3)
labels batch shape: (32,)
```

Identifying a small convolution for dogs vs. cats categories

In [8]:

```
from tensorflow import keras
from tensorflow.keras import layers

input_1000 = keras.Input(shape=(180, 180, 3))
d_1000 = layers.Rescaling(1./255)(input_1000)
d_1000 = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(d_1000)
d_1000 = layers.MaxPooling2D(pool_size=2)(d_1000)
d_1000 = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(d_1000)
d_1000 = layers.MaxPooling2D(pool_size=2)(d_1000)
d_1000 = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(d_1000)
d_1000 = layers.MaxPooling2D(pool_size=2)(d_1000)
d_1000 = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(d_1000)
d_1000 = layers.MaxPooling2D(pool_size=2)(d_1000)
d_1000 = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(d_1000)
d_1000 = layers.Flatten()(d_1000)
d_1000 = layers.Dropout(0.5)(d_1000)
output_1000 = layers.Dense(1, activation="sigmoid")(d_1000)
model_1000 = keras.Model(inputs=input_1000, outputs=output_1000)
```

Model Training

In [9]:

```
model_1000.compile(loss="binary_crossentropy",
optimizer="adam",
metrics=["accuracy"])
```

The training dataset is used to train the model after it has been built. We use the validation dataset to verify the model's performance at the end of each epoch. I'm utilizing GPU to reduce the time it takes for each epoch to execute

In [10]:

```
model_1000.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_1 (InputLayer)        [(None, 180, 180, 3)]     0         
                                                                 
 rescaling (Rescaling)       (None, 180, 180, 3)       0         
                                                                 
 conv2d (Conv2D)             (None, 178, 178, 32)      896       
                                                                 
 max_pooling2d (MaxPooling2  (None, 89, 89, 32)        0         
 D)                                                              
                                                                 
 conv2d_1 (Conv2D)           (None, 87, 87, 64)        18496     
                                                                 
 max_pooling2d_1 (MaxPoolin  (None, 43, 43, 64)        0         
 g2D)                                                            
                                                                 
 conv2d_2 (Conv2D)           (None, 41, 41, 128)       73856     
                                                                 
 max_pooling2d_2 (MaxPoolin  (None, 20, 20, 128)       0         
 g2D)                                                            
                                                                 
 conv2d_3 (Conv2D)           (None, 18, 18, 256)       295168    
                                                                 
 max_pooling2d_3 (MaxPoolin  (None, 9, 9, 256)         0         
 g2D)                                                            
                                                                 
 conv2d_4 (Conv2D)           (None, 7, 7, 256)         590080    
                                                                 
 flatten (Flatten)           (None, 12544)             0         
                                                                 
 dropout (Dropout)           (None, 12544)             0         
                                                                 
 dense (Dense)               (None, 1)                 12545     
                                                                 
=================================================================
Total params: 991041 (3.78 MB)
Trainable params: 991041 (3.78 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Model Fitting

In [11]:

```
callback_1000 = [keras.callbacks.ModelCheckpoint(filepath="convnet_from_scratch.ker

history_1000 = model_1000.fit(train_data,epochs=100,validation_data=valid_data,call
Epoch 1/100
63/63 [==============================] - 17s 135ms/step - loss: 0.6977 - accuracy:
0.5015 - val_loss: 0.6910 - val_accuracy: 0.5000
Epoch 2/100
63/63 [==============================] - 5s 71ms/step - loss: 0.6911 - accuracy:
0.5150 - val_loss: 0.6876 - val_accuracy: 0.5040
Epoch 3/100
63/63 [==============================] - 6s 82ms/step - loss: 0.6644 - accuracy:
0.5955 - val_loss: 0.6468 - val_accuracy: 0.6310
Epoch 4/100
63/63 [==============================] - 6s 84ms/step - loss: 0.6382 - accuracy:
0.6355 - val_loss: 0.6558 - val_accuracy: 0.6220
Epoch 5/100
63/63 [==============================] - 9s 124ms/step - loss: 0.6093 - accuracy:
0.6675 - val_loss: 0.6169 - val_accuracy: 0.6600
Epoch 6/100
63/63 [==============================] - 6s 86ms/step - loss: 0.5644 - accuracy:
0.7005 - val_loss: 0.6193 - val_accuracy: 0.6850
Epoch 7/100
63/63 [==============================] - 5s 76ms/step - loss: 0.5418 - accuracy:
0.7405 - val_loss: 0.6278 - val_accuracy: 0.7000
Epoch 8/100
63/63 [==============================] - 4s 60ms/step - loss: 0.5000 - accuracy:
0.7565 - val_loss: 0.6093 - val_accuracy: 0.6960
Epoch 9/100
63/63 [==============================] - 4s 60ms/step - loss: 0.4554 - accuracy:
0.7865 - val_loss: 0.6203 - val_accuracy: 0.7090
Epoch 10/100
63/63 [==============================] - 9s 130ms/step - loss: 0.4145 - accuracy:
0.8050 - val_loss: 0.6349 - val_accuracy: 0.7150
Epoch 11/100
63/63 [==============================] - 4s 56ms/step - loss: 0.3649 - accuracy:
0.8290 - val_loss: 0.7276 - val_accuracy: 0.7170
Epoch 12/100
63/63 [==============================] - 6s 85ms/step - loss: 0.3275 - accuracy:
0.8505 - val_loss: 0.7079 - val_accuracy: 0.7120
Epoch 13/100
63/63 [==============================] - 7s 96ms/step - loss: 0.2919 - accuracy:
0.8755 - val_loss: 0.6604 - val_accuracy: 0.7360
Epoch 14/100
63/63 [==============================] - 11s 164ms/step - loss: 0.2131 - accuracy:
0.9115 - val_loss: 0.7788 - val_accuracy: 0.7250
Epoch 15/100
63/63 [==============================] - 7s 107ms/step - loss: 0.1922 - accuracy:
0.9200 - val_loss: 0.7700 - val_accuracy: 0.7240
Epoch 16/100
63/63 [==============================] - 5s 81ms/step - loss: 0.1473 - accuracy:
0.9465 - val_loss: 0.8657 - val_accuracy: 0.7320
Epoch 17/100
63/63 [==============================] - 9s 136ms/step - loss: 0.1298 - accuracy:
0.9465 - val_loss: 1.0012 - val_accuracy: 0.7340
Epoch 18/100
63/63 [==============================] - 6s 95ms/step - loss: 0.1348 - accuracy:
0.9505 - val_loss: 0.9659 - val_accuracy: 0.7370
Epoch 19/100
63/63 [==============================] - 7s 97ms/step - loss: 0.0878 - accuracy:
```

```
0.9675 - val_loss: 0.8854 - val_accuracy: 0.7630
Epoch 20/100
63/63 [==============================] - 6s 97ms/step - loss: 0.0770 - accuracy:
0.9745 - val_loss: 1.0033 - val_accuracy: 0.7530
Epoch 21/100
63/63 [==============================] - 8s 122ms/step - loss: 0.0500 - accuracy:
0.9820 - val_loss: 1.0940 - val_accuracy: 0.7740
Epoch 22/100
63/63 [==============================] - 5s 77ms/step - loss: 0.0638 - accuracy:
0.9825 - val_loss: 0.9676 - val_accuracy: 0.7400
Epoch 23/100
63/63 [==============================] - 5s 79ms/step - loss: 0.0715 - accuracy:
0.9775 - val_loss: 1.0264 - val_accuracy: 0.7450
Epoch 24/100
63/63 [==============================] - 4s 56ms/step - loss: 0.0543 - accuracy:
0.9800 - val_loss: 1.0621 - val_accuracy: 0.7600
Epoch 25/100
63/63 [==============================] - 8s 120ms/step - loss: 0.0288 - accuracy:
0.9910 - val_loss: 1.3484 - val_accuracy: 0.7620
Epoch 26/100
63/63 [==============================] - 6s 73ms/step - loss: 0.0282 - accuracy:
0.9910 - val_loss: 1.3228 - val_accuracy: 0.7480
Epoch 27/100
63/63 [==============================] - 10s 141ms/step - loss: 0.0352 - accuracy:
0.9880 - val_loss: 1.4085 - val_accuracy: 0.7560
Epoch 28/100
63/63 [==============================] - 6s 89ms/step - loss: 0.0433 - accuracy:
0.9845 - val_loss: 1.2298 - val_accuracy: 0.7490
Epoch 29/100
63/63 [==============================] - 6s 84ms/step - loss: 0.0473 - accuracy:
0.9820 - val_loss: 1.2861 - val_accuracy: 0.7520
Epoch 30/100
63/63 [==============================] - 4s 56ms/step - loss: 0.0311 - accuracy:
0.9900 - val_loss: 1.3625 - val_accuracy: 0.7520
Epoch 31/100
63/63 [==============================] - 5s 83ms/step - loss: 0.0292 - accuracy:
0.9915 - val_loss: 1.3803 - val_accuracy: 0.7480
Epoch 32/100
63/63 [==============================] - 6s 79ms/step - loss: 0.0209 - accuracy:
0.9935 - val_loss: 1.5386 - val_accuracy: 0.7560
Epoch 33/100
63/63 [==============================] - 5s 70ms/step - loss: 0.0311 - accuracy:
0.9945 - val_loss: 1.3950 - val_accuracy: 0.7640
Epoch 34/100
63/63 [==============================] - 7s 111ms/step - loss: 0.0305 - accuracy:
0.9885 - val_loss: 1.5966 - val_accuracy: 0.7490
Epoch 35/100
63/63 [==============================] - 5s 72ms/step - loss: 0.0256 - accuracy:
0.9910 - val_loss: 1.6983 - val_accuracy: 0.7260
Epoch 36/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0763 - accuracy:
0.9780 - val_loss: 1.2404 - val_accuracy: 0.7500
Epoch 37/100
63/63 [==============================] - 12s 179ms/step - loss: 0.0195 - accuracy:
0.9935 - val_loss: 1.2723 - val_accuracy: 0.7520
Epoch 38/100
```

```
63/63 [==============================] - 8s 116ms/step - loss: 0.0138 - accuracy:
0.9945 - val_loss: 1.3169 - val_accuracy: 0.7650
Epoch 39/100
63/63 [==============================] - 4s 57ms/step - loss: 0.0094 - accuracy:
0.9955 - val_loss: 1.5010 - val_accuracy: 0.7610
Epoch 40/100
63/63 [==============================] - 6s 87ms/step - loss: 0.0058 - accuracy:
0.9990 - val_loss: 1.8244 - val_accuracy: 0.7590
Epoch 41/100
63/63 [==============================] - 8s 109ms/step - loss: 0.0254 - accuracy:
0.9900 - val_loss: 1.6613 - val_accuracy: 0.7450
Epoch 42/100
63/63 [==============================] - 5s 81ms/step - loss: 0.0383 - accuracy:
0.9825 - val_loss: 1.5890 - val_accuracy: 0.7440
Epoch 43/100
63/63 [==============================] - 6s 92ms/step - loss: 0.0237 - accuracy:
0.9935 - val_loss: 1.4276 - val_accuracy: 0.7640
Epoch 44/100
63/63 [==============================] - 7s 108ms/step - loss: 0.0294 - accuracy:
0.9905 - val_loss: 1.7531 - val_accuracy: 0.7370
Epoch 45/100
63/63 [==============================] - 7s 109ms/step - loss: 0.0643 - accuracy:
0.9800 - val_loss: 1.6110 - val_accuracy: 0.7320
Epoch 46/100
63/63 [==============================] - 7s 108ms/step - loss: 0.0328 - accuracy:
0.9900 - val_loss: 1.6124 - val_accuracy: 0.7360
Epoch 47/100
63/63 [==============================] - 5s 73ms/step - loss: 0.0226 - accuracy:
0.9920 - val_loss: 1.5882 - val_accuracy: 0.7500
Epoch 48/100
63/63 [==============================] - 7s 101ms/step - loss: 0.0267 - accuracy:
0.9875 - val_loss: 1.9205 - val_accuracy: 0.7460
Epoch 49/100
63/63 [==============================] - 5s 76ms/step - loss: 0.0599 - accuracy:
0.9780 - val_loss: 1.3641 - val_accuracy: 0.7310
Epoch 50/100
63/63 [==============================] - 6s 94ms/step - loss: 0.0117 - accuracy:
0.9955 - val_loss: 1.6092 - val_accuracy: 0.7650
Epoch 51/100
63/63 [==============================] - 6s 82ms/step - loss: 0.0166 - accuracy:
0.9950 - val_loss: 1.4059 - val_accuracy: 0.7600
Epoch 52/100
63/63 [==============================] - 8s 129ms/step - loss: 0.0206 - accuracy:
0.9915 - val_loss: 1.6251 - val_accuracy: 0.7540
Epoch 53/100
63/63 [==============================] - 8s 113ms/step - loss: 0.0186 - accuracy:
0.9935 - val_loss: 1.7538 - val_accuracy: 0.7510
Epoch 54/100
63/63 [==============================] - 8s 117ms/step - loss: 0.0245 - accuracy:
0.9935 - val_loss: 1.3937 - val_accuracy: 0.7630
Epoch 55/100
63/63 [==============================] - 5s 68ms/step - loss: 0.0754 - accuracy:
0.9785 - val_loss: 1.2327 - val_accuracy: 0.7490
Epoch 56/100
63/63 [==============================] - 5s 78ms/step - loss: 0.0557 - accuracy:
0.9790 - val_loss: 1.6350 - val_accuracy: 0.7370
```

```
Epoch 57/100
63/63 [==============================] - 6s 98ms/step - loss: 0.0188 - accuracy:
0.9935 - val_loss: 1.6471 - val_accuracy: 0.7340
Epoch 58/100
63/63 [==============================] - 4s 56ms/step - loss: 0.0154 - accuracy:
0.9920 - val_loss: 1.9929 - val_accuracy: 0.7410
Epoch 59/100
63/63 [==============================] - 6s 88ms/step - loss: 0.0240 - accuracy:
0.9930 - val_loss: 1.8422 - val_accuracy: 0.7470
Epoch 60/100
63/63 [==============================] - 5s 73ms/step - loss: 0.0158 - accuracy:
0.9935 - val_loss: 1.8816 - val_accuracy: 0.7460
Epoch 61/100
63/63 [==============================] - 4s 57ms/step - loss: 0.0061 - accuracy:
0.9985 - val_loss: 1.7865 - val_accuracy: 0.7620
Epoch 62/100
63/63 [==============================] - 8s 129ms/step - loss: 0.0026 - accuracy:
0.9990 - val_loss: 1.9099 - val_accuracy: 0.7660
Epoch 63/100
63/63 [==============================] - 7s 103ms/step - loss: 0.0104 - accuracy:
0.9975 - val_loss: 1.7052 - val_accuracy: 0.7760
Epoch 64/100
63/63 [==============================] - 6s 98ms/step - loss: 0.0189 - accuracy:
0.9925 - val_loss: 1.9192 - val_accuracy: 0.7580
Epoch 65/100
63/63 [==============================] - 6s 77ms/step - loss: 0.0177 - accuracy:
0.9960 - val_loss: 2.0375 - val_accuracy: 0.7560
Epoch 66/100
63/63 [==============================] - 10s 154ms/step - loss: 0.0080 - accuracy:
0.9970 - val_loss: 1.6548 - val_accuracy: 0.7630
Epoch 67/100
63/63 [==============================] - 7s 110ms/step - loss: 0.0174 - accuracy:
0.9950 - val_loss: 2.1316 - val_accuracy: 0.7220
Epoch 68/100
63/63 [==============================] - 8s 130ms/step - loss: 0.0357 - accuracy:
0.9870 - val_loss: 1.6922 - val_accuracy: 0.7540
Epoch 69/100
63/63 [==============================] - 7s 105ms/step - loss: 0.0527 - accuracy:
0.9825 - val_loss: 1.6951 - val_accuracy: 0.7540
Epoch 70/100
63/63 [==============================] - 7s 106ms/step - loss: 0.0158 - accuracy:
0.9940 - val_loss: 2.0583 - val_accuracy: 0.7500
Epoch 71/100
63/63 [==============================] - 4s 57ms/step - loss: 0.0352 - accuracy:
0.9915 - val_loss: 2.0175 - val_accuracy: 0.7490
Epoch 72/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0136 - accuracy:
0.9940 - val_loss: 1.7755 - val_accuracy: 0.7440
Epoch 73/100
63/63 [==============================] - 6s 98ms/step - loss: 0.0211 - accuracy:
0.9945 - val_loss: 1.8644 - val_accuracy: 0.7680
Epoch 74/100
63/63 [==============================] - 5s 72ms/step - loss: 0.0084 - accuracy:
0.9965 - val_loss: 2.3228 - val_accuracy: 0.7440
Epoch 75/100
63/63 [==============================] - 7s 115ms/step - loss: 0.0253 - accuracy:
```

```
0.9920 - val_loss: 2.2015 - val_accuracy: 0.7310
Epoch 76/100
63/63 [==============================] - 6s 87ms/step - loss: 0.0222 - accuracy:
0.9925 - val_loss: 1.8939 - val_accuracy: 0.7440
Epoch 77/100
63/63 [==============================] - 6s 96ms/step - loss: 0.0168 - accuracy:
0.9945 - val_loss: 1.9867 - val_accuracy: 0.7600
Epoch 78/100
63/63 [==============================] - 4s 57ms/step - loss: 0.0321 - accuracy:
0.9895 - val_loss: 2.1752 - val_accuracy: 0.7270
Epoch 79/100
63/63 [==============================] - 7s 105ms/step - loss: 0.0285 - accuracy:
0.9920 - val_loss: 1.5438 - val_accuracy: 0.7600
Epoch 80/100
63/63 [==============================] - 5s 69ms/step - loss: 0.0121 - accuracy:
0.9965 - val_loss: 2.8209 - val_accuracy: 0.7200
Epoch 81/100
63/63 [==============================] - 9s 131ms/step - loss: 0.0219 - accuracy:
0.9910 - val_loss: 2.1963 - val_accuracy: 0.7370
Epoch 82/100
63/63 [==============================] - 7s 97ms/step - loss: 0.0384 - accuracy:
0.9835 - val_loss: 2.1566 - val_accuracy: 0.7360
Epoch 83/100
63/63 [==============================] - 5s 79ms/step - loss: 0.0351 - accuracy:
0.9860 - val_loss: 2.0909 - val_accuracy: 0.7430
Epoch 84/100
63/63 [==============================] - 12s 186ms/step - loss: 0.0364 - accuracy:
0.9885 - val_loss: 2.4829 - val_accuracy: 0.7310
Epoch 85/100
63/63 [==============================] - 4s 57ms/step - loss: 0.0460 - accuracy:
0.9855 - val_loss: 2.1655 - val_accuracy: 0.7350
Epoch 86/100
63/63 [==============================] - 4s 65ms/step - loss: 0.0416 - accuracy:
0.9905 - val_loss: 1.6923 - val_accuracy: 0.7600
Epoch 87/100
63/63 [==============================] - 6s 95ms/step - loss: 0.0128 - accuracy:
0.9965 - val_loss: 1.7706 - val_accuracy: 0.7600
Epoch 88/100
63/63 [==============================] - 4s 56ms/step - loss: 0.0047 - accuracy:
0.9980 - val_loss: 1.9456 - val_accuracy: 0.7590
Epoch 89/100
63/63 [==============================] - 4s 56ms/step - loss: 0.0019 - accuracy:
0.9995 - val_loss: 1.8100 - val_accuracy: 0.7610
Epoch 90/100
63/63 [==============================] - 5s 76ms/step - loss: 9.9616e-04 - accurac
y: 1.0000 - val_loss: 1.9284 - val_accuracy: 0.7590
Epoch 91/100
63/63 [==============================] - 6s 80ms/step - loss: 0.0089 - accuracy:
0.9975 - val_loss: 1.9015 - val_accuracy: 0.7700
Epoch 92/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0114 - accuracy:
0.9975 - val_loss: 1.9445 - val_accuracy: 0.7620
Epoch 93/100
63/63 [==============================] - 11s 172ms/step - loss: 0.0104 - accuracy:
0.9950 - val_loss: 2.5797 - val_accuracy: 0.7530
Epoch 94/100
```

```
63/63 [==============================] - 5s 64ms/step - loss: 0.0050 - accuracy:
0.9975 - val_loss: 2.1600 - val_accuracy: 0.7590
Epoch 95/100
63/63 [==============================] - 6s 92ms/step - loss: 0.0240 - accuracy:
0.9930 - val_loss: 2.3961 - val_accuracy: 0.7440
Epoch 96/100
63/63 [==============================] - 4s 66ms/step - loss: 0.0396 - accuracy:
0.9895 - val_loss: 1.9169 - val_accuracy: 0.7390
Epoch 97/100
63/63 [==============================] - 6s 89ms/step - loss: 0.0361 - accuracy:
0.9890 - val_loss: 1.6473 - val_accuracy: 0.7560
Epoch 98/100
63/63 [==============================] - 4s 57ms/step - loss: 0.0082 - accuracy:
0.9965 - val_loss: 1.6212 - val_accuracy: 0.7620
Epoch 99/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0161 - accuracy:
0.9945 - val_loss: 2.2999 - val_accuracy: 0.7520
Epoch 100/100
63/63 [==============================] - 6s 86ms/step - loss: 0.0258 - accuracy:
0.9910 - val_loss: 1.8844 - val_accuracy: 0.7640
```

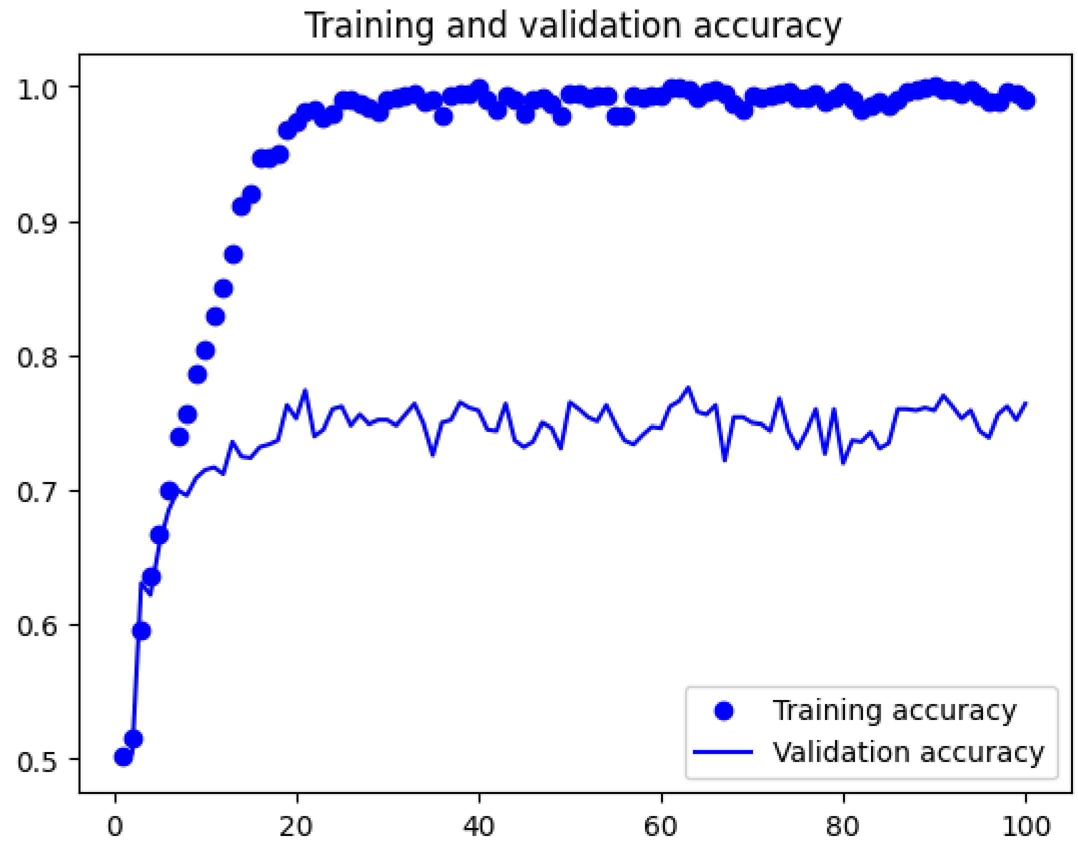
Plot for loss and accuracy during training

In [12]:

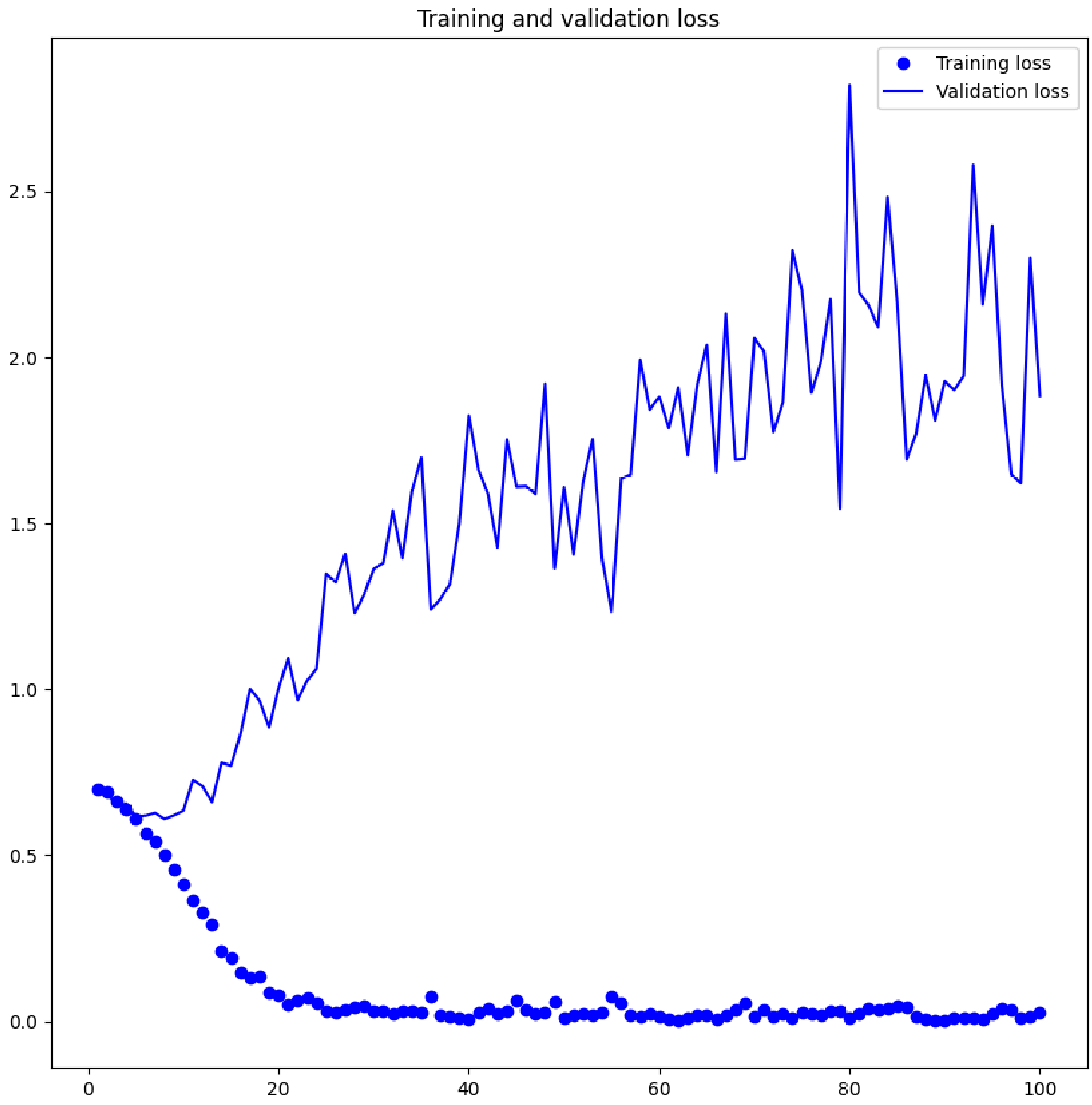
```python
import matplotlib.pyplot as plt

accuracy = history_1000.history["accuracy"]
val_accuracy = history_1000.history["val_accuracy"]
loss = history_1000.history["loss"]
val_loss = history_1000.history["val_loss"]
epochs = range(1, len(accuracy) + 1)
plt.plot(epochs, accuracy, "bo", label="Training accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.legend()
plt.figure()

plt.plot(epochs, loss, "bo", label="Training loss")
plt.plot(epochs, val_loss, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.legend()
plt.show()
```


Training and validation accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0
20
40
60
80
100
● Training accuracy
Validation accuracy

```
<Figure size 640x480 with 0 Axes>
```



Test Accuracy of the model

In [13]:

```
test_1000 = keras.models.load_model("convnet_from_scratch.keras")
test_loss, test_acc = test_1000.evaluate(test_data)
print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [==============================] - 2s 34ms/step - loss: 0.5793 - accuracy:
0.7210
Test accuracy: 0.721
```

Q2. Increase your training sample size. You may pick any amount. Keep the validation and test samples the same as above. Optimize your network (again training from scratch). What performance did you achieve?

Using data augmentation

In [14]:

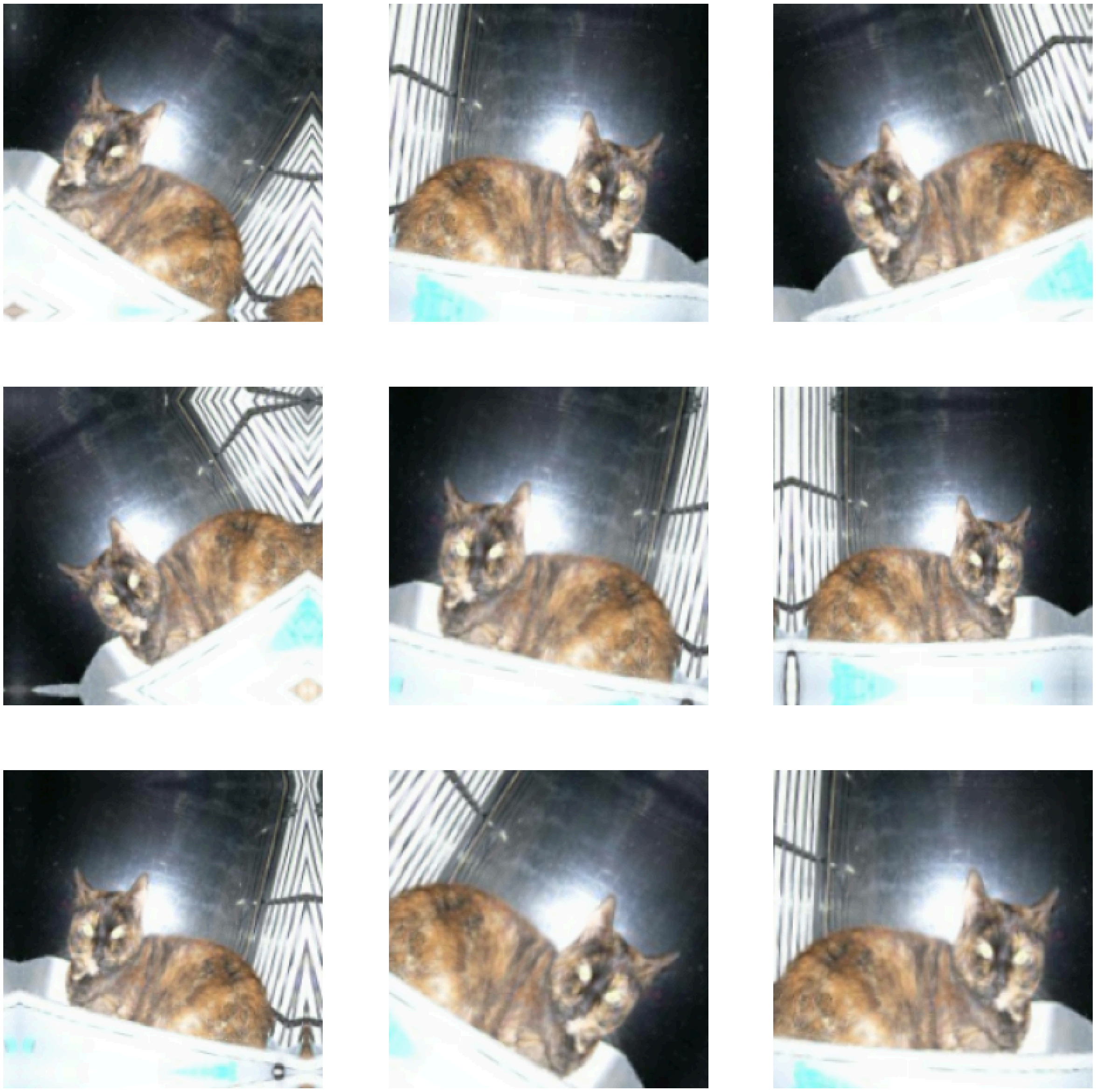
```
shutil.rmtree("./cats_vs_dogs_small_Q2", ignore_errors=True)
o_dir = pathlib.Path("train")
n_dir = pathlib.Path("cats_vs_dogs_small_Q2")
def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = n_dir / subset_name / category
        os.makedirs(dir)
        fnames = [f"{category}.{i}.jpg" for i in range(start_index, end_index)]
        for fname in fnames:
            shutil.copyfile(src=o_dir / fname,
            dst=dir / fname)

make_subset("train", start_index=500, end_index=2000)
make_subset("validation", start_index=2000, end_index=2500)
make_subset("test", start_index=2500, end_index=3000)

augmentation = keras.Sequential([layers.RandomFlip("horizontal"),layers.RandomRotat

plt.figure(figsize=(10, 10))

for images, _ in train_data.take(1):
    for i in range(9):
        augmented_img= augmentation(images)
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(augmented_img[0].numpy().astype("uint8"))
        plt.axis("off")
```



Convolutional neural network with dropout and picture augmentation

In [15]:

```
input_1500 = keras.Input(shape=(180, 180, 3))

d_2000 = augmentation(input_1500)
d_2000 = layers.Rescaling(1./255)(d_2000)
d_2000 = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(d_2000)
d_2000 = layers.MaxPooling2D(pool_size=2)(d_2000)
d_2000 = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(d_2000)
d_2000 = layers.MaxPooling2D(pool_size=2)(d_2000)
d_2000 = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(d_2000)
d_2000 = layers.MaxPooling2D(pool_size=2)(d_2000)
d_2000 = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(d_2000)
d_2000 = layers.MaxPooling2D(pool_size=2)(d_2000)
d_2000 = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(d_2000)
d_2000 = layers.Flatten()(d_2000)
d_2000 = layers.Dropout(0.5)(d_2000)

output_1500 = layers.Dense(1, activation="sigmoid")(d_2000)
```

```
model_1500 = keras.Model(inputs=input_1500, outputs=output_1500)

model_1500.compile(loss="binary_crossentropy",optimizer="adam",metrics=["accuracy"]

callback_1500 = [keras.callbacks.ModelCheckpoint(filepath="convnet_from_scratch_wit

history_1500 = model_1500.fit(train_data,epochs=100,validation_data=valid_data,call
Epoch 1/100
63/63 [==============================] - 9s 97ms/step - loss: 0.6948 - accuracy:
0.5035 - val_loss: 0.6923 - val_accuracy: 0.5000
Epoch 2/100
63/63 [==============================] - 6s 95ms/step - loss: 0.6924 - accuracy:
0.5065 - val_loss: 0.8598 - val_accuracy: 0.5000
Epoch 3/100
63/63 [==============================] - 4s 64ms/step - loss: 0.6969 - accuracy:
0.5120 - val_loss: 0.6946 - val_accuracy: 0.5000
Epoch 4/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6939 - accuracy:
0.5075 - val_loss: 0.6926 - val_accuracy: 0.5010
Epoch 5/100
63/63 [==============================] - 7s 112ms/step - loss: 0.6939 - accuracy:
0.5060 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 6/100
63/63 [==============================] - 4s 60ms/step - loss: 0.6934 - accuracy:
0.5010 - val_loss: 0.6929 - val_accuracy: 0.5000
Epoch 7/100
63/63 [==============================] - 6s 88ms/step - loss: 0.6923 - accuracy:
0.5200 - val_loss: 0.6940 - val_accuracy: 0.5000
Epoch 8/100
63/63 [==============================] - 7s 102ms/step - loss: 0.6867 - accuracy:
0.5625 - val_loss: 0.7015 - val_accuracy: 0.5090
Epoch 9/100
63/63 [==============================] - 4s 60ms/step - loss: 0.6787 - accuracy:
0.5800 - val_loss: 0.6812 - val_accuracy: 0.5520
Epoch 10/100
63/63 [==============================] - 10s 150ms/step - loss: 0.6666 - accuracy:
0.5945 - val_loss: 0.6714 - val_accuracy: 0.5900
Epoch 11/100
63/63 [==============================] - 7s 95ms/step - loss: 0.6485 - accuracy:
0.6345 - val_loss: 0.6641 - val_accuracy: 0.6200
Epoch 12/100
63/63 [==============================] - 5s 77ms/step - loss: 0.6488 - accuracy:
0.6280 - val_loss: 0.6783 - val_accuracy: 0.6010
Epoch 13/100
63/63 [==============================] - 9s 130ms/step - loss: 0.6322 - accuracy:
0.6455 - val_loss: 0.6889 - val_accuracy: 0.6120
Epoch 14/100
63/63 [==============================] - 7s 110ms/step - loss: 0.6310 - accuracy:
0.6475 - val_loss: 0.7608 - val_accuracy: 0.5560
Epoch 15/100
63/63 [==============================] - 9s 145ms/step - loss: 0.6458 - accuracy:
0.6360 - val_loss: 0.6432 - val_accuracy: 0.6610
Epoch 16/100
63/63 [==============================] - 6s 93ms/step - loss: 0.6269 - accuracy:
0.6575 - val_loss: 0.6282 - val_accuracy: 0.6740
Epoch 17/100
63/63 [==============================] - 10s 156ms/step - loss: 0.6205 - accuracy:
0.6650 - val_loss: 0.6263 - val_accuracy: 0.6810
Epoch 18/100
63/63 [==============================] - 6s 88ms/step - loss: 0.6009 - accuracy:
0.6775 - val_loss: 0.7148 - val_accuracy: 0.5430
Epoch 19/100
63/63 [==============================] - 11s 162ms/step - loss: 0.6120 - accuracy:
```

```
0.6685 - val_loss: 0.6101 - val_accuracy: 0.6800
Epoch 20/100
63/63 [==============================] - 7s 104ms/step - loss: 0.5909 - accuracy:
0.6985 - val_loss: 0.6487 - val_accuracy: 0.6310
Epoch 21/100
63/63 [==============================] - 4s 61ms/step - loss: 0.5836 - accuracy:
0.6945 - val_loss: 0.5876 - val_accuracy: 0.7030
Epoch 22/100
63/63 [==============================] - 4s 66ms/step - loss: 0.5806 - accuracy:
0.6900 - val_loss: 0.5795 - val_accuracy: 0.7320
Epoch 23/100
63/63 [==============================] - 9s 137ms/step - loss: 0.5853 - accuracy:
0.6975 - val_loss: 0.5672 - val_accuracy: 0.7290
Epoch 24/100
63/63 [==============================] - 7s 98ms/step - loss: 0.5449 - accuracy:
0.7325 - val_loss: 0.5760 - val_accuracy: 0.7160
Epoch 25/100
63/63 [==============================] - 6s 97ms/step - loss: 0.5586 - accuracy:
0.7155 - val_loss: 0.5740 - val_accuracy: 0.7270
Epoch 26/100
63/63 [==============================] - 7s 101ms/step - loss: 0.5412 - accuracy:
0.7320 - val_loss: 0.5507 - val_accuracy: 0.7210
Epoch 27/100
63/63 [==============================] - 8s 116ms/step - loss: 0.5197 - accuracy:
0.7465 - val_loss: 0.5344 - val_accuracy: 0.7460
Epoch 28/100
63/63 [==============================] - 8s 103ms/step - loss: 0.5127 - accuracy:
0.7485 - val_loss: 0.5760 - val_accuracy: 0.7260
Epoch 29/100
63/63 [==============================] - 4s 58ms/step - loss: 0.4953 - accuracy:
0.7545 - val_loss: 0.5722 - val_accuracy: 0.7420
Epoch 30/100
63/63 [==============================] - 7s 103ms/step - loss: 0.5043 - accuracy:
0.7535 - val_loss: 0.6184 - val_accuracy: 0.7080
Epoch 31/100
63/63 [==============================] - 5s 77ms/step - loss: 0.4869 - accuracy:
0.7660 - val_loss: 0.5450 - val_accuracy: 0.7420
Epoch 32/100
63/63 [==============================] - 8s 127ms/step - loss: 0.4794 - accuracy:
0.7660 - val_loss: 0.5306 - val_accuracy: 0.7550
Epoch 33/100
63/63 [==============================] - 5s 76ms/step - loss: 0.4540 - accuracy:
0.7915 - val_loss: 0.5411 - val_accuracy: 0.7490
Epoch 34/100
63/63 [==============================] - 4s 57ms/step - loss: 0.4466 - accuracy:
0.7830 - val_loss: 0.5701 - val_accuracy: 0.7490
Epoch 35/100
63/63 [==============================] - 9s 137ms/step - loss: 0.4466 - accuracy:
0.7935 - val_loss: 0.5476 - val_accuracy: 0.7380
Epoch 36/100
63/63 [==============================] - 5s 80ms/step - loss: 0.4504 - accuracy:
0.7825 - val_loss: 0.5336 - val_accuracy: 0.7720
Epoch 37/100
63/63 [==============================] - 5s 83ms/step - loss: 0.4191 - accuracy:
0.8060 - val_loss: 0.5332 - val_accuracy: 0.7790
Epoch 38/100
```

```
63/63 [==============================] - 5s 80ms/step - loss: 0.4099 - accuracy:
0.8055 - val_loss: 0.5190 - val_accuracy: 0.7740
Epoch 39/100
63/63 [==============================] - 4s 58ms/step - loss: 0.4146 - accuracy:
0.8190 - val_loss: 0.5341 - val_accuracy: 0.7750
Epoch 40/100
63/63 [==============================] - 9s 135ms/step - loss: 0.4172 - accuracy:
0.8025 - val_loss: 0.5109 - val_accuracy: 0.7860
Epoch 41/100
63/63 [==============================] - 6s 93ms/step - loss: 0.3819 - accuracy:
0.8335 - val_loss: 0.4883 - val_accuracy: 0.7910
Epoch 42/100
63/63 [==============================] - 5s 64ms/step - loss: 0.3839 - accuracy:
0.8300 - val_loss: 0.4803 - val_accuracy: 0.7820
Epoch 43/100
63/63 [==============================] - 4s 56ms/step - loss: 0.3782 - accuracy:
0.8345 - val_loss: 0.5331 - val_accuracy: 0.7590
Epoch 44/100
63/63 [==============================] - 6s 98ms/step - loss: 0.3675 - accuracy:
0.8365 - val_loss: 0.5139 - val_accuracy: 0.7920
Epoch 45/100
63/63 [==============================] - 4s 57ms/step - loss: 0.3617 - accuracy:
0.8350 - val_loss: 0.5978 - val_accuracy: 0.7690
Epoch 46/100
63/63 [==============================] - 4s 63ms/step - loss: 0.3709 - accuracy:
0.8320 - val_loss: 0.4917 - val_accuracy: 0.8010
Epoch 47/100
63/63 [==============================] - 7s 100ms/step - loss: 0.3524 - accuracy:
0.8455 - val_loss: 0.4814 - val_accuracy: 0.7950
Epoch 48/100
63/63 [==============================] - 4s 58ms/step - loss: 0.3522 - accuracy:
0.8330 - val_loss: 0.5023 - val_accuracy: 0.8020
Epoch 49/100
63/63 [==============================] - 4s 59ms/step - loss: 0.3341 - accuracy:
0.8490 - val_loss: 0.4869 - val_accuracy: 0.7860
Epoch 50/100
63/63 [==============================] - 6s 98ms/step - loss: 0.3370 - accuracy:
0.8555 - val_loss: 0.4815 - val_accuracy: 0.7990
Epoch 51/100
63/63 [==============================] - 5s 71ms/step - loss: 0.3228 - accuracy:
0.8530 - val_loss: 0.5013 - val_accuracy: 0.8040
Epoch 52/100
63/63 [==============================] - 4s 58ms/step - loss: 0.3081 - accuracy:
0.8700 - val_loss: 0.5622 - val_accuracy: 0.7950
Epoch 53/100
63/63 [==============================] - 5s 71ms/step - loss: 0.3063 - accuracy:
0.8700 - val_loss: 0.5699 - val_accuracy: 0.7850
Epoch 54/100
63/63 [==============================] - 6s 94ms/step - loss: 0.3100 - accuracy:
0.8635 - val_loss: 0.5057 - val_accuracy: 0.8000
Epoch 55/100
63/63 [==============================] - 4s 58ms/step - loss: 0.3019 - accuracy:
0.8650 - val_loss: 0.5081 - val_accuracy: 0.7960
Epoch 56/100
63/63 [==============================] - 4s 59ms/step - loss: 0.2793 - accuracy:
0.8815 - val_loss: 0.5186 - val_accuracy: 0.7950
```

```
Epoch 57/100
63/63 [==============================] - 5s 81ms/step - loss: 0.2955 - accuracy:
0.8695 - val_loss: 0.5772 - val_accuracy: 0.7900
Epoch 58/100
63/63 [==============================] - 4s 57ms/step - loss: 0.2902 - accuracy:
0.8775 - val_loss: 0.4967 - val_accuracy: 0.7940
Epoch 59/100
63/63 [==============================] - 6s 90ms/step - loss: 0.2766 - accuracy:
0.8850 - val_loss: 0.5385 - val_accuracy: 0.8020
Epoch 60/100
63/63 [==============================] - 7s 97ms/step - loss: 0.2704 - accuracy:
0.8840 - val_loss: 0.4849 - val_accuracy: 0.8210
Epoch 61/100
63/63 [==============================] - 4s 58ms/step - loss: 0.2797 - accuracy:
0.8835 - val_loss: 0.4881 - val_accuracy: 0.7970
Epoch 62/100
63/63 [==============================] - 9s 145ms/step - loss: 0.2730 - accuracy:
0.8820 - val_loss: 0.5432 - val_accuracy: 0.8040
Epoch 63/100
63/63 [==============================] - 6s 89ms/step - loss: 0.2767 - accuracy:
0.8810 - val_loss: 0.5151 - val_accuracy: 0.8010
Epoch 64/100
63/63 [==============================] - 6s 94ms/step - loss: 0.2353 - accuracy:
0.8955 - val_loss: 0.7731 - val_accuracy: 0.7710
Epoch 65/100
63/63 [==============================] - 8s 118ms/step - loss: 0.2923 - accuracy:
0.8740 - val_loss: 0.4833 - val_accuracy: 0.8110
Epoch 66/100
63/63 [==============================] - 6s 90ms/step - loss: 0.2370 - accuracy:
0.9045 - val_loss: 0.5736 - val_accuracy: 0.8030
Epoch 67/100
63/63 [==============================] - 5s 64ms/step - loss: 0.2529 - accuracy:
0.8910 - val_loss: 0.5306 - val_accuracy: 0.7990
Epoch 68/100
63/63 [==============================] - 4s 60ms/step - loss: 0.2786 - accuracy:
0.8780 - val_loss: 0.5656 - val_accuracy: 0.8230
Epoch 69/100
63/63 [==============================] - 5s 73ms/step - loss: 0.2465 - accuracy:
0.8960 - val_loss: 0.5350 - val_accuracy: 0.8160
Epoch 70/100
63/63 [==============================] - 8s 118ms/step - loss: 0.2336 - accuracy:
0.9030 - val_loss: 0.5139 - val_accuracy: 0.8120
Epoch 71/100
63/63 [==============================] - 5s 79ms/step - loss: 0.2203 - accuracy:
0.9025 - val_loss: 0.5161 - val_accuracy: 0.8200
Epoch 72/100
63/63 [==============================] - 9s 131ms/step - loss: 0.2165 - accuracy:
0.9140 - val_loss: 0.6121 - val_accuracy: 0.7890
Epoch 73/100
63/63 [==============================] - 5s 84ms/step - loss: 0.2114 - accuracy:
0.9140 - val_loss: 0.6113 - val_accuracy: 0.8030
Epoch 74/100
63/63 [==============================] - 5s 72ms/step - loss: 0.2275 - accuracy:
0.9105 - val_loss: 0.5463 - val_accuracy: 0.8100
Epoch 75/100
63/63 [==============================] - 9s 124ms/step - loss: 0.2355 - accuracy:
```

```
0.9050 - val_loss: 0.5467 - val_accuracy: 0.8270
Epoch 76/100
63/63 [==============================] - 4s 60ms/step - loss: 0.2233 - accuracy:
0.9090 - val_loss: 0.5732 - val_accuracy: 0.8190
Epoch 77/100
63/63 [==============================] - 5s 77ms/step - loss: 0.2171 - accuracy:
0.9100 - val_loss: 0.5115 - val_accuracy: 0.8070
Epoch 78/100
63/63 [==============================] - 9s 140ms/step - loss: 0.2016 - accuracy:
0.9120 - val_loss: 0.5276 - val_accuracy: 0.8270
Epoch 79/100
63/63 [==============================] - 6s 98ms/step - loss: 0.1955 - accuracy:
0.9215 - val_loss: 0.6222 - val_accuracy: 0.8140
Epoch 80/100
63/63 [==============================] - 9s 134ms/step - loss: 0.2111 - accuracy:
0.9110 - val_loss: 0.5540 - val_accuracy: 0.8250
Epoch 81/100
63/63 [==============================] - 4s 60ms/step - loss: 0.2042 - accuracy:
0.9090 - val_loss: 0.5584 - val_accuracy: 0.8120
Epoch 82/100
63/63 [==============================] - 8s 122ms/step - loss: 0.2253 - accuracy:
0.9105 - val_loss: 0.6160 - val_accuracy: 0.7860
Epoch 83/100
63/63 [==============================] - 7s 107ms/step - loss: 0.1859 - accuracy:
0.9275 - val_loss: 0.5418 - val_accuracy: 0.8340
Epoch 84/100
63/63 [==============================] - 6s 96ms/step - loss: 0.1798 - accuracy:
0.9275 - val_loss: 0.5827 - val_accuracy: 0.8240
Epoch 85/100
63/63 [==============================] - 4s 61ms/step - loss: 0.1963 - accuracy:
0.9215 - val_loss: 0.5600 - val_accuracy: 0.8090
Epoch 86/100
63/63 [==============================] - 5s 77ms/step - loss: 0.1916 - accuracy:
0.9150 - val_loss: 0.5849 - val_accuracy: 0.8150
Epoch 87/100
63/63 [==============================] - 9s 143ms/step - loss: 0.2030 - accuracy:
0.9135 - val_loss: 0.5653 - val_accuracy: 0.8250
Epoch 88/100
63/63 [==============================] - 6s 96ms/step - loss: 0.1824 - accuracy:
0.9220 - val_loss: 0.5697 - val_accuracy: 0.8220
Epoch 89/100
63/63 [==============================] - 9s 136ms/step - loss: 0.1753 - accuracy:
0.9240 - val_loss: 0.5939 - val_accuracy: 0.8230
Epoch 90/100
63/63 [==============================] - 8s 106ms/step - loss: 0.1739 - accuracy:
0.9335 - val_loss: 0.6005 - val_accuracy: 0.8260
Epoch 91/100
63/63 [==============================] - 10s 159ms/step - loss: 0.1812 - accuracy:
0.9260 - val_loss: 0.5746 - val_accuracy: 0.8170
Epoch 92/100
63/63 [==============================] - 6s 89ms/step - loss: 0.1608 - accuracy:
0.9305 - val_loss: 0.5664 - val_accuracy: 0.8250
Epoch 93/100
63/63 [==============================] - 6s 93ms/step - loss: 0.1840 - accuracy:
0.9240 - val_loss: 0.5595 - val_accuracy: 0.8220
Epoch 94/100
```

```
63/63 [==============================] - 5s 66ms/step - loss: 0.2135 - accuracy:
0.9165 - val_loss: 0.6017 - val_accuracy: 0.8180
Epoch 95/100
63/63 [==============================] - 9s 134ms/step - loss: 0.1561 - accuracy:
0.9365 - val_loss: 0.5939 - val_accuracy: 0.8080
Epoch 96/100
63/63 [==============================] - 5s 81ms/step - loss: 0.1807 - accuracy:
0.9240 - val_loss: 0.5941 - val_accuracy: 0.8110
Epoch 97/100
63/63 [==============================] - 6s 98ms/step - loss: 0.1773 - accuracy:
0.9295 - val_loss: 0.6170 - val_accuracy: 0.8140
Epoch 98/100
63/63 [==============================] - 4s 58ms/step - loss: 0.1635 - accuracy:
0.9365 - val_loss: 0.6153 - val_accuracy: 0.8180
Epoch 99/100
63/63 [==============================] - 5s 74ms/step - loss: 0.1554 - accuracy:
0.9380 - val_loss: 0.6059 - val_accuracy: 0.8250
Epoch 100/100
63/63 [==============================] - 9s 132ms/step - loss: 0.1576 - accuracy:
0.9420 - val_loss: 0.6843 - val_accuracy: 0.8150
```

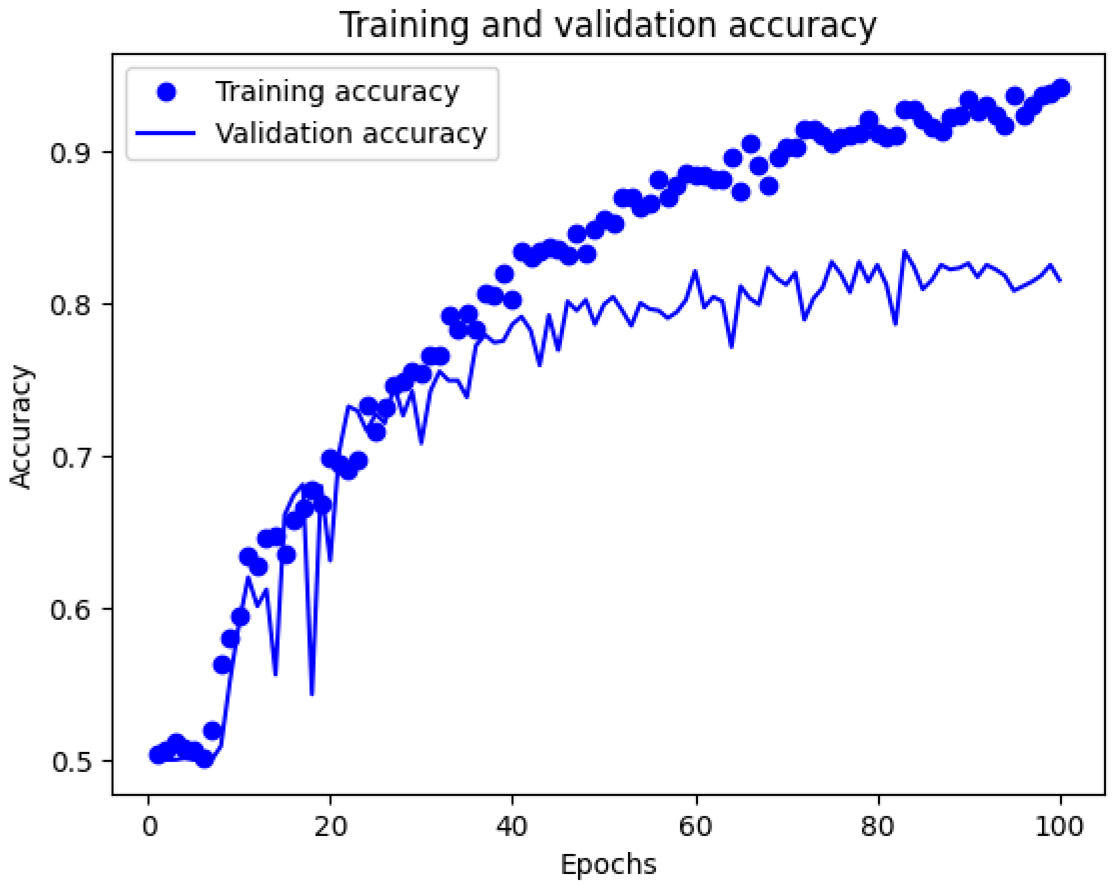
Plot for loss and accuracy during training

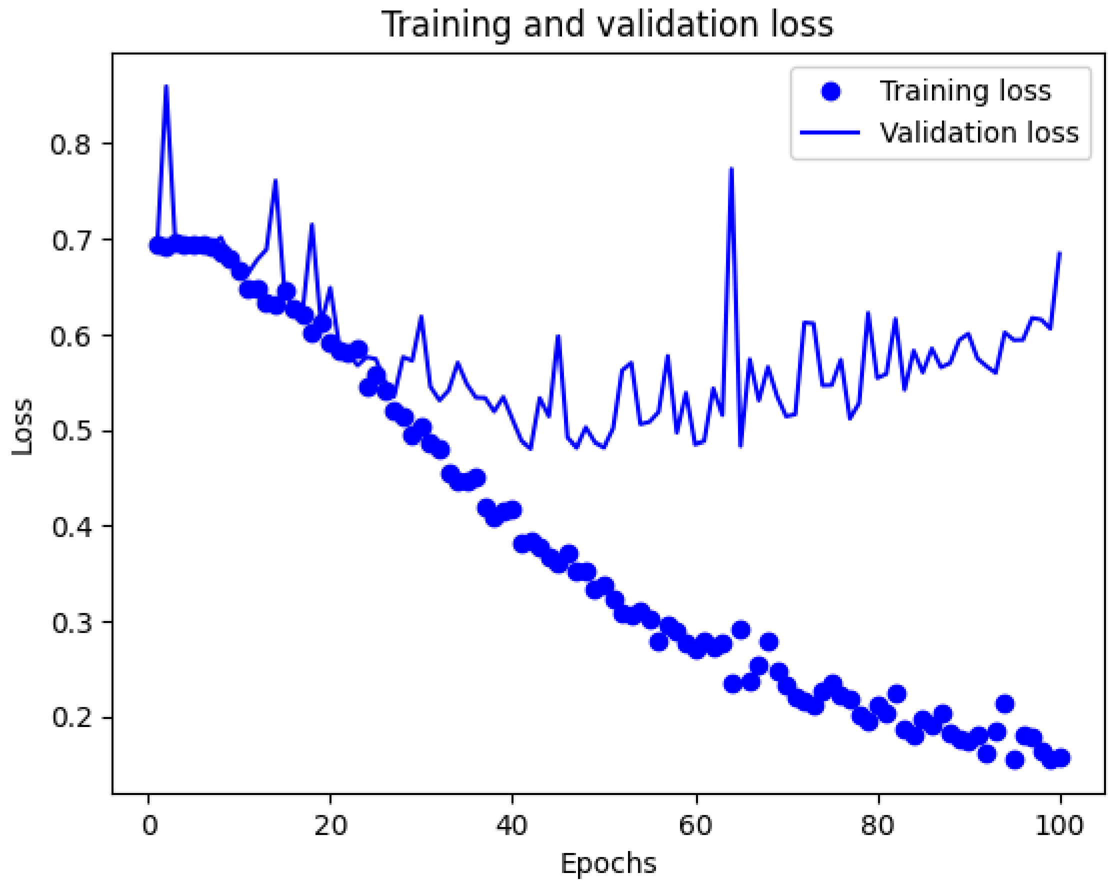
In [17]:

```python
accuracy_1500 = history_1500.history["accuracy"]
valac_1500 = history_1500.history["val_accuracy"]
loss_1500 = history_1500.history["loss"]
valloss_1500 = history_1500.history["val_loss"]
epochs = range(1, len(accuracy_1500) + 1)

plt.plot(epochs, accuracy_1500, "bo", label="Training accuracy")
plt.plot(epochs, valac_1500, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()

plt.figure()
plt.plot(epochs, loss_1500, "bo", label="Training loss")
plt.plot(epochs, valloss_1500, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```


Training and validation accuracy


Training and validation loss

Test Accuracy of model

In [18]:

```
test_1500 = keras.models.load_model("convnet_from_scratch_with_augmentation_info.ke

test_loss, test_acc = test_1500.evaluate(test_data)

print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [==============================] - 1s 29ms/step - loss: 0.4281 - accuracy:
0.8030
Test accuracy: 0.803
```

Q3. Now change your training sample so that you achieve better performance than those from Steps 1 and 2. This sample size may be larger, or smaller than those in the previous steps. The objective is to find the ideal training sample size to get best prediction results.

Increasing the training sample to 2000, keeping the Validation and test sets the same as before(500 samples)

In [19]:

```
n_dir = pathlib.Path("cats_vs_dogs_small_Q3")

def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = n_dir / subset_name / category
        os.makedirs(dir, exist_ok = True)
        fnames = [f"{category}.{i}.jpg" for i in range(start_index, end_index)]
        for fname in fnames:
            shutil.copyfile(src=o_dir / fname,
            dst=dir / fname)

make_subset("train", start_index=500, end_index=2500)
make_subset("validation", start_index=2500, end_index=3000)
make_subset("test", start_index=3000, end_index=3500)

input_2000 = keras.Input(shape=(180, 180, 3))

d_2000 = augmentation_info(input_2000)
d_2000 = layers.Rescaling(1./255)(d_2000)
d_2000 = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(d_2000)
d_2000 = layers.MaxPooling2D(pool_size=2)(d_2000)
d_2000 = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(d_2000)
d_2000 = layers.MaxPooling2D(pool_size=2)(d_2000)
d_2000 = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(d_2000)
d_2000 = layers.MaxPooling2D(pool_size=2)(d_2000)
d_2000 = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(d_2000)
d_2000 = layers.MaxPooling2D(pool_size=2)(d_2000)
d_2000 = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(d_2000)
d_2000 = layers.Flatten()(d_2000)
d_2000 = layers.Dropout(0.5)(d_2000)

output_2000 = layers.Dense(1, activation="sigmoid")(d_2000)
```

```
model_2000 = keras.Model(inputs=input_2000, outputs=output_2000)

model_2000.compile(loss="binary_crossentropy",optimizer="adam",metrics=["accuracy"]

callback_2000 = [keras.callbacks.ModelCheckpoint(filepath="convnet_from_scratch_wit

history_2000 = model_2000.fit(train_data,epochs=100,validation_data=valid_data,call
Epoch 1/100
63/63 [==============================] - 8s 92ms/step - loss: 0.6943 - accuracy:
0.4970 - val_loss: 0.6922 - val_accuracy: 0.5040
Epoch 2/100
63/63 [==============================] - 4s 58ms/step - loss: 0.6932 - accuracy:
0.5200 - val_loss: 0.7187 - val_accuracy: 0.5030
Epoch 3/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6910 - accuracy:
0.5340 - val_loss: 0.6901 - val_accuracy: 0.5160
Epoch 4/100
63/63 [==============================] - 6s 93ms/step - loss: 0.6786 - accuracy:
0.5665 - val_loss: 0.8541 - val_accuracy: 0.5180
Epoch 5/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6841 - accuracy:
0.5565 - val_loss: 0.6790 - val_accuracy: 0.5730
Epoch 6/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6697 - accuracy:
0.5895 - val_loss: 0.6770 - val_accuracy: 0.5750
Epoch 7/100
63/63 [==============================] - 7s 107ms/step - loss: 0.6724 - accuracy:
0.5915 - val_loss: 0.6710 - val_accuracy: 0.5910
Epoch 8/100
63/63 [==============================] - 4s 58ms/step - loss: 0.6585 - accuracy:
0.6245 - val_loss: 0.6860 - val_accuracy: 0.5880
Epoch 9/100
63/63 [==============================] - 4s 57ms/step - loss: 0.6464 - accuracy:
0.6255 - val_loss: 0.7003 - val_accuracy: 0.6050
Epoch 10/100
63/63 [==============================] - 7s 111ms/step - loss: 0.6434 - accuracy:
0.6370 - val_loss: 0.6818 - val_accuracy: 0.5850
Epoch 11/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6318 - accuracy:
0.6415 - val_loss: 0.6452 - val_accuracy: 0.6340
Epoch 12/100
63/63 [==============================] - 4s 58ms/step - loss: 0.6175 - accuracy:
0.6650 - val_loss: 0.6302 - val_accuracy: 0.6470
Epoch 13/100
63/63 [==============================] - 9s 141ms/step - loss: 0.6166 - accuracy:
0.6595 - val_loss: 0.6397 - val_accuracy: 0.6380
Epoch 14/100
63/63 [==============================] - 4s 58ms/step - loss: 0.5992 - accuracy:
0.6880 - val_loss: 0.6220 - val_accuracy: 0.6530
Epoch 15/100
63/63 [==============================] - 4s 58ms/step - loss: 0.6046 - accuracy:
0.6680 - val_loss: 0.6282 - val_accuracy: 0.6340
Epoch 16/100
63/63 [==============================] - 6s 98ms/step - loss: 0.5978 - accuracy:
0.6735 - val_loss: 0.6039 - val_accuracy: 0.6570
Epoch 17/100
63/63 [==============================] - 4s 58ms/step - loss: 0.5779 - accuracy:
0.7060 - val_loss: 0.6505 - val_accuracy: 0.6520
Epoch 18/100
63/63 [==============================] - 4s 66ms/step - loss: 0.5786 - accuracy:
0.6955 - val_loss: 0.5679 - val_accuracy: 0.7040
Epoch 19/100
63/63 [==============================] - 6s 86ms/step - loss: 0.5497 - accuracy:
```

```
0.7130 - val_loss: 0.6493 - val_accuracy: 0.6760
Epoch 20/100
63/63 [==============================] - 4s 57ms/step - loss: 0.5389 - accuracy:
0.7090 - val_loss: 0.5742 - val_accuracy: 0.7050
Epoch 21/100
63/63 [==============================] - 4s 57ms/step - loss: 0.5511 - accuracy:
0.7200 - val_loss: 0.6435 - val_accuracy: 0.6610
Epoch 22/100
63/63 [==============================] - 6s 90ms/step - loss: 0.5250 - accuracy:
0.7340 - val_loss: 0.6085 - val_accuracy: 0.6890
Epoch 23/100
63/63 [==============================] - 4s 58ms/step - loss: 0.5158 - accuracy:
0.7290 - val_loss: 0.5990 - val_accuracy: 0.7060
Epoch 24/100
63/63 [==============================] - 6s 86ms/step - loss: 0.5111 - accuracy:
0.7450 - val_loss: 0.5771 - val_accuracy: 0.7050
Epoch 25/100
63/63 [==============================] - 5s 75ms/step - loss: 0.5094 - accuracy:
0.7480 - val_loss: 0.5430 - val_accuracy: 0.7380
Epoch 26/100
63/63 [==============================] - 4s 59ms/step - loss: 0.4982 - accuracy:
0.7570 - val_loss: 0.5662 - val_accuracy: 0.7150
Epoch 27/100
63/63 [==============================] - 4s 64ms/step - loss: 0.4973 - accuracy:
0.7560 - val_loss: 0.5408 - val_accuracy: 0.7290
Epoch 28/100
63/63 [==============================] - 6s 87ms/step - loss: 0.4721 - accuracy:
0.7665 - val_loss: 0.5596 - val_accuracy: 0.7210
Epoch 29/100
63/63 [==============================] - 4s 57ms/step - loss: 0.4575 - accuracy:
0.7855 - val_loss: 0.5570 - val_accuracy: 0.7450
Epoch 30/100
63/63 [==============================] - 4s 58ms/step - loss: 0.4761 - accuracy:
0.7700 - val_loss: 0.5599 - val_accuracy: 0.7420
Epoch 31/100
63/63 [==============================] - 6s 96ms/step - loss: 0.4407 - accuracy:
0.7930 - val_loss: 0.5223 - val_accuracy: 0.7460
Epoch 32/100
63/63 [==============================] - 5s 68ms/step - loss: 0.4349 - accuracy:
0.7960 - val_loss: 0.5477 - val_accuracy: 0.7400
Epoch 33/100
63/63 [==============================] - 4s 56ms/step - loss: 0.4196 - accuracy:
0.8050 - val_loss: 0.5695 - val_accuracy: 0.7310
Epoch 34/100
63/63 [==============================] - 4s 62ms/step - loss: 0.4414 - accuracy:
0.8020 - val_loss: 0.5452 - val_accuracy: 0.7660
Epoch 35/100
63/63 [==============================] - 7s 98ms/step - loss: 0.4121 - accuracy:
0.8070 - val_loss: 0.5426 - val_accuracy: 0.7430
Epoch 36/100
63/63 [==============================] - 4s 58ms/step - loss: 0.4082 - accuracy:
0.8120 - val_loss: 0.5552 - val_accuracy: 0.7390
Epoch 37/100
63/63 [==============================] - 4s 59ms/step - loss: 0.3824 - accuracy:
0.8345 - val_loss: 0.5739 - val_accuracy: 0.7560
Epoch 38/100
```

```
63/63 [==============================] - 5s 77ms/step - loss: 0.4045 - accuracy:
0.8100 - val_loss: 0.5476 - val_accuracy: 0.7860
Epoch 39/100
63/63 [==============================] - 4s 59ms/step - loss: 0.3781 - accuracy:
0.8255 - val_loss: 0.5630 - val_accuracy: 0.7780
Epoch 40/100
63/63 [==============================] - 4s 61ms/step - loss: 0.3889 - accuracy:
0.8235 - val_loss: 0.5696 - val_accuracy: 0.7770
Epoch 41/100
63/63 [==============================] - 7s 111ms/step - loss: 0.3777 - accuracy:
0.8240 - val_loss: 0.6148 - val_accuracy: 0.7600
Epoch 42/100
63/63 [==============================] - 4s 66ms/step - loss: 0.3807 - accuracy:
0.8290 - val_loss: 0.5206 - val_accuracy: 0.7730
Epoch 43/100
63/63 [==============================] - 4s 59ms/step - loss: 0.3593 - accuracy:
0.8430 - val_loss: 0.5566 - val_accuracy: 0.7570
Epoch 44/100
63/63 [==============================] - 6s 97ms/step - loss: 0.3525 - accuracy:
0.8440 - val_loss: 0.5963 - val_accuracy: 0.7620
Epoch 45/100
63/63 [==============================] - 5s 69ms/step - loss: 0.3380 - accuracy:
0.8425 - val_loss: 0.5750 - val_accuracy: 0.7750
Epoch 46/100
63/63 [==============================] - 4s 59ms/step - loss: 0.3459 - accuracy:
0.8580 - val_loss: 0.5376 - val_accuracy: 0.7980
Epoch 47/100
63/63 [==============================] - 5s 74ms/step - loss: 0.3344 - accuracy:
0.8545 - val_loss: 0.5632 - val_accuracy: 0.7730
Epoch 48/100
63/63 [==============================] - 6s 86ms/step - loss: 0.3115 - accuracy:
0.8580 - val_loss: 0.6148 - val_accuracy: 0.7730
Epoch 49/100
63/63 [==============================] - 4s 58ms/step - loss: 0.3388 - accuracy:
0.8505 - val_loss: 0.5457 - val_accuracy: 0.7740
Epoch 50/100
63/63 [==============================] - 6s 94ms/step - loss: 0.3337 - accuracy:
0.8595 - val_loss: 0.5340 - val_accuracy: 0.7860
Epoch 51/100
63/63 [==============================] - 4s 58ms/step - loss: 0.3226 - accuracy:
0.8590 - val_loss: 0.5562 - val_accuracy: 0.7830
Epoch 52/100
63/63 [==============================] - 7s 108ms/step - loss: 0.3329 - accuracy:
0.8545 - val_loss: 0.5699 - val_accuracy: 0.7770
Epoch 53/100
63/63 [==============================] - 4s 58ms/step - loss: 0.3055 - accuracy:
0.8615 - val_loss: 0.5675 - val_accuracy: 0.8040
Epoch 54/100
63/63 [==============================] - 4s 57ms/step - loss: 0.3170 - accuracy:
0.8575 - val_loss: 0.5235 - val_accuracy: 0.7930
Epoch 55/100
63/63 [==============================] - 6s 90ms/step - loss: 0.3164 - accuracy:
0.8675 - val_loss: 0.5673 - val_accuracy: 0.7880
Epoch 56/100
63/63 [==============================] - 4s 57ms/step - loss: 0.3027 - accuracy:
0.8580 - val_loss: 0.5709 - val_accuracy: 0.7920
```

```
Epoch 57/100
63/63 [==============================] - 6s 89ms/step - loss: 0.2949 - accuracy:
0.8725 - val_loss: 0.5471 - val_accuracy: 0.7870
Epoch 58/100
63/63 [==============================] - 4s 58ms/step - loss: 0.3043 - accuracy:
0.8665 - val_loss: 0.5763 - val_accuracy: 0.7820
Epoch 59/100
63/63 [==============================] - 4s 63ms/step - loss: 0.2557 - accuracy:
0.8925 - val_loss: 0.5896 - val_accuracy: 0.7930
Epoch 60/100
63/63 [==============================] - 6s 88ms/step - loss: 0.3138 - accuracy:
0.8635 - val_loss: 0.5785 - val_accuracy: 0.7860
Epoch 61/100
63/63 [==============================] - 4s 57ms/step - loss: 0.2885 - accuracy:
0.8685 - val_loss: 0.5984 - val_accuracy: 0.7920
Epoch 62/100
63/63 [==============================] - 4s 60ms/step - loss: 0.2561 - accuracy:
0.8940 - val_loss: 0.6758 - val_accuracy: 0.7710
Epoch 63/100
63/63 [==============================] - 5s 82ms/step - loss: 0.2734 - accuracy:
0.8735 - val_loss: 0.6469 - val_accuracy: 0.7740
Epoch 64/100
63/63 [==============================] - 5s 77ms/step - loss: 0.2554 - accuracy:
0.8880 - val_loss: 0.5983 - val_accuracy: 0.7980
Epoch 65/100
63/63 [==============================] - 4s 58ms/step - loss: 0.2253 - accuracy:
0.9110 - val_loss: 0.7112 - val_accuracy: 0.7730
Epoch 66/100
63/63 [==============================] - 4s 59ms/step - loss: 0.2576 - accuracy:
0.8920 - val_loss: 0.5552 - val_accuracy: 0.8120
Epoch 67/100
63/63 [==============================] - 7s 107ms/step - loss: 0.2440 - accuracy:
0.8950 - val_loss: 0.6794 - val_accuracy: 0.7770
Epoch 68/100
63/63 [==============================] - 4s 59ms/step - loss: 0.2359 - accuracy:
0.9010 - val_loss: 0.5944 - val_accuracy: 0.7950
Epoch 69/100
63/63 [==============================] - 4s 57ms/step - loss: 0.2442 - accuracy:
0.8910 - val_loss: 0.6617 - val_accuracy: 0.8050
Epoch 70/100
63/63 [==============================] - 6s 97ms/step - loss: 0.2248 - accuracy:
0.9040 - val_loss: 0.6453 - val_accuracy: 0.8020
Epoch 71/100
63/63 [==============================] - 4s 58ms/step - loss: 0.2029 - accuracy:
0.9130 - val_loss: 0.6759 - val_accuracy: 0.8070
Epoch 72/100
63/63 [==============================] - 5s 82ms/step - loss: 0.2492 - accuracy:
0.8900 - val_loss: 0.6017 - val_accuracy: 0.8190
Epoch 73/100
63/63 [==============================] - 6s 86ms/step - loss: 0.2237 - accuracy:
0.9095 - val_loss: 0.6425 - val_accuracy: 0.8010
Epoch 74/100
63/63 [==============================] - 4s 57ms/step - loss: 0.2143 - accuracy:
0.9125 - val_loss: 0.6429 - val_accuracy: 0.8080
Epoch 75/100
63/63 [==============================] - 4s 57ms/step - loss: 0.1939 - accuracy:
```

```
0.9215 - val_loss: 0.7058 - val_accuracy: 0.7930
Epoch 76/100
63/63 [==============================] - 6s 97ms/step - loss: 0.2347 - accuracy:
0.9060 - val_loss: 0.6490 - val_accuracy: 0.8090
Epoch 77/100
63/63 [==============================] - 5s 64ms/step - loss: 0.2143 - accuracy:
0.9145 - val_loss: 0.6350 - val_accuracy: 0.7980
Epoch 78/100
63/63 [==============================] - 4s 58ms/step - loss: 0.2007 - accuracy:
0.9085 - val_loss: 0.6828 - val_accuracy: 0.8090
Epoch 79/100
63/63 [==============================] - 5s 73ms/step - loss: 0.1920 - accuracy:
0.9215 - val_loss: 0.7029 - val_accuracy: 0.8010
Epoch 80/100
63/63 [==============================] - 6s 86ms/step - loss: 0.1820 - accuracy:
0.9220 - val_loss: 0.6238 - val_accuracy: 0.8070
Epoch 81/100
63/63 [==============================] - 4s 60ms/step - loss: 0.1971 - accuracy:
0.9220 - val_loss: 0.6576 - val_accuracy: 0.8040
Epoch 82/100
63/63 [==============================] - 4s 63ms/step - loss: 0.1706 - accuracy:
0.9280 - val_loss: 0.7515 - val_accuracy: 0.7990
Epoch 83/100
63/63 [==============================] - 7s 108ms/step - loss: 0.2122 - accuracy:
0.9125 - val_loss: 0.6097 - val_accuracy: 0.7950
Epoch 84/100
63/63 [==============================] - 4s 57ms/step - loss: 0.1794 - accuracy:
0.9245 - val_loss: 0.7206 - val_accuracy: 0.7970
Epoch 85/100
63/63 [==============================] - 4s 58ms/step - loss: 0.1854 - accuracy:
0.9240 - val_loss: 0.7020 - val_accuracy: 0.8060
Epoch 86/100
63/63 [==============================] - 6s 97ms/step - loss: 0.1825 - accuracy:
0.9215 - val_loss: 0.7863 - val_accuracy: 0.7900
Epoch 87/100
63/63 [==============================] - 4s 58ms/step - loss: 0.1550 - accuracy:
0.9410 - val_loss: 0.7305 - val_accuracy: 0.7980
Epoch 88/100
63/63 [==============================] - 5s 82ms/step - loss: 0.1800 - accuracy:
0.9240 - val_loss: 0.7703 - val_accuracy: 0.8050
Epoch 89/100
63/63 [==============================] - 6s 90ms/step - loss: 0.1773 - accuracy:
0.9230 - val_loss: 0.6649 - val_accuracy: 0.8190
Epoch 90/100
63/63 [==============================] - 4s 64ms/step - loss: 0.1618 - accuracy:
0.9365 - val_loss: 0.7373 - val_accuracy: 0.8240
Epoch 91/100
63/63 [==============================] - 4s 60ms/step - loss: 0.1763 - accuracy:
0.9360 - val_loss: 0.7640 - val_accuracy: 0.8120
Epoch 92/100
63/63 [==============================] - 7s 111ms/step - loss: 0.1775 - accuracy:
0.9280 - val_loss: 0.6101 - val_accuracy: 0.8010
Epoch 93/100
63/63 [==============================] - 4s 62ms/step - loss: 0.1675 - accuracy:
0.9345 - val_loss: 0.8550 - val_accuracy: 0.7870
Epoch 94/100
```

```
63/63 [==============================] - 5s 75ms/step - loss: 0.1671 - accuracy:
0.9330 - val_loss: 0.6772 - val_accuracy: 0.8100
Epoch 95/100
63/63 [==============================] - 7s 98ms/step - loss: 0.1655 - accuracy:
0.9310 - val_loss: 0.7333 - val_accuracy: 0.7910
Epoch 96/100
63/63 [==============================] - 4s 61ms/step - loss: 0.1918 - accuracy:
0.9265 - val_loss: 0.6701 - val_accuracy: 0.7990
Epoch 97/100
63/63 [==============================] - 4s 64ms/step - loss: 0.1562 - accuracy:
0.9345 - val_loss: 0.7408 - val_accuracy: 0.8200
Epoch 98/100
63/63 [==============================] - 7s 101ms/step - loss: 0.1366 - accuracy:
0.9445 - val_loss: 0.8378 - val_accuracy: 0.8110
Epoch 99/100
63/63 [==============================] - 4s 60ms/step - loss: 0.1535 - accuracy:
0.9345 - val_loss: 0.7448 - val_accuracy: 0.8220
Epoch 100/100
63/63 [==============================] - 4s 60ms/step - loss: 0.1485 - accuracy:
0.9410 - val_loss: 0.8130 - val_accuracy: 0.8030
```

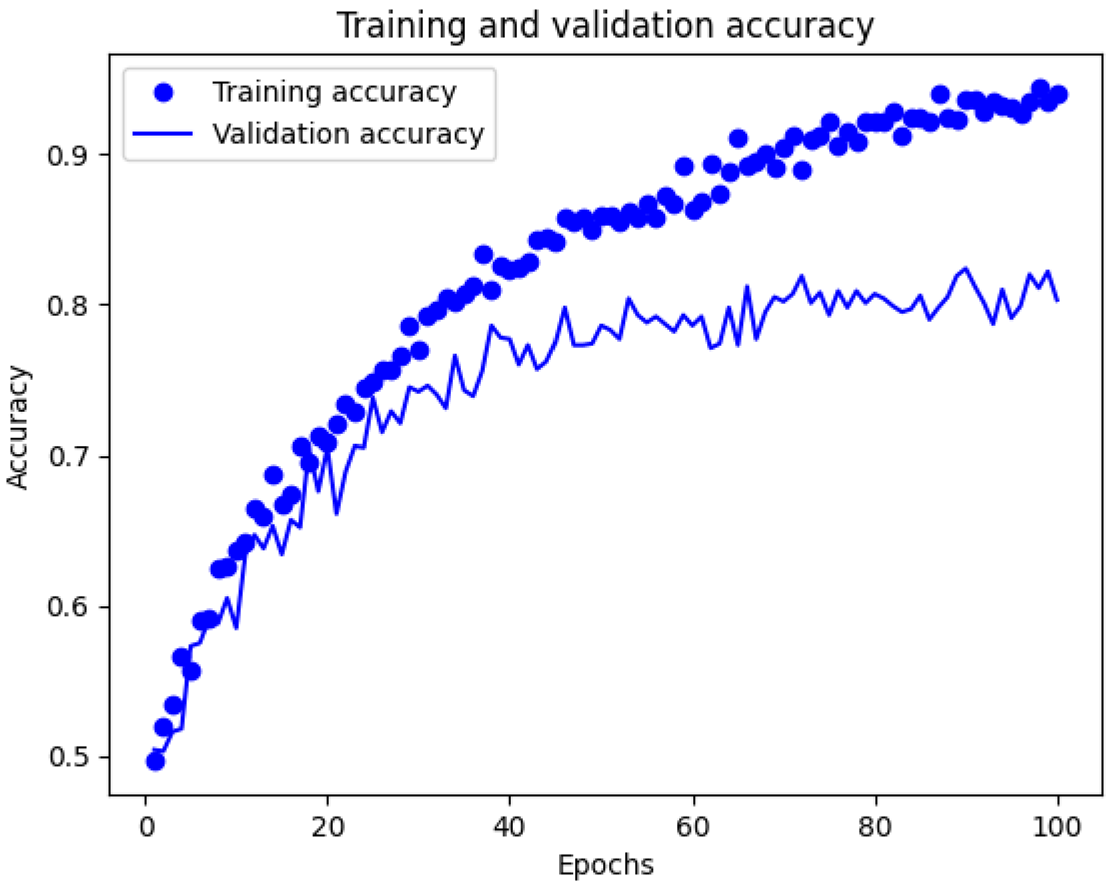
Plot for loss and accuracy during training

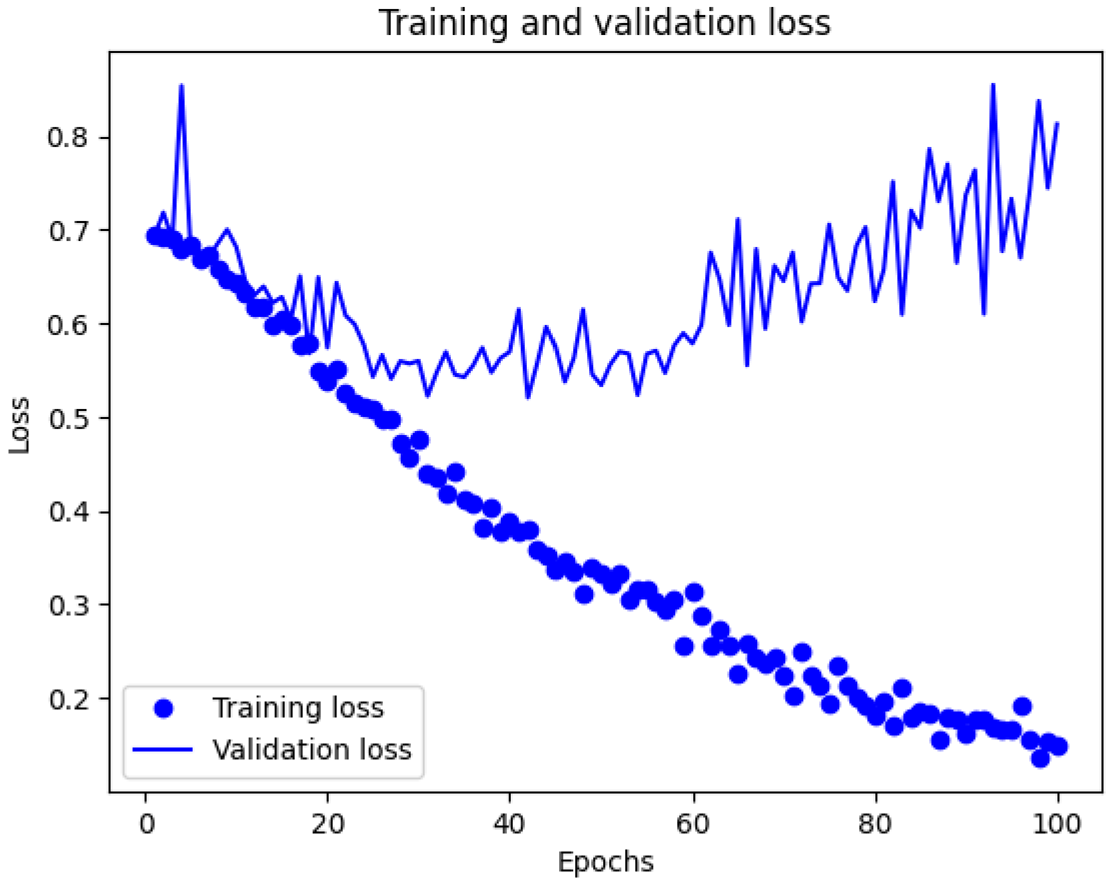
In [21]:

```
accuracy_2000 = history_2000.history["accuracy"]
valac_2000 = history_2000.history["val_accuracy"]
loss_2000 = history_2000.history["loss"]
valloss_2000 = history_2000.history["val_loss"]
epochs = range(1, len(accuracy_2000) + 1)

plt.plot(epochs, accuracy_2000, "bo", label="Training accuracy")
plt.plot(epochs, valac_2000, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()

plt.figure()
plt.plot(epochs, loss_2000, "bo", label="Training loss")
plt.plot(epochs, valloss_2000, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```


Training and validation accuracy
Training accuracy
Validation accuracy
Accuracy
0.5
0.6
0.7
0.8
0.9
0
20
40
60
80
100
Epochs


Training and validation loss
Training loss
Validation loss
Loss
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
20
40
60
80
100
Epochs

Test Accuracy of the model

In [22]:

```
test_2000 = keras.models.load_model("convnet_from_scratch_with_augmentation_info.ke

test_loss, test_acc = test_2000.evaluate(test_data)

print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [==============================] - 1s 28ms/step - loss: 0.4650 - accuracy:
0.7920
Test accuracy: 0.792
```

Q4. Repeat Steps 1-3, but now using a pretrained network. The sample sizes you use in Steps 2 and 3 for the pretrained network may be the same or different from those using the network where you trained from scratch. Again, use any and all optimization techniques to get best performance.

Instantiating the VGG16 convolutional base

In [23]:

```
convoluted_b = keras.applications.vgg16.VGG16(weights="imagenet",include_top=False,
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-application
s/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [==============================] - 0s 0us/step
```

In [24]:

```
convoluted_b.summary()
```

```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 180, 180, 3)]     0

 block1_conv1 (Conv2D)       (None, 180, 180, 64)      1792

 block1_conv2 (Conv2D)       (None, 180, 180, 64)      36928

 block1_pool (MaxPooling2D)  (None, 90, 90, 64)        0

 block2_conv1 (Conv2D)       (None, 90, 90, 128)       73856

 block2_conv2 (Conv2D)       (None, 90, 90, 128)       147584

 block2_pool (MaxPooling2D)  (None, 45, 45, 128)       0

 block3_conv1 (Conv2D)       (None, 45, 45, 256)       295168

 block3_conv2 (Conv2D)       (None, 45, 45, 256)       590080

 block3_conv3 (Conv2D)       (None, 45, 45, 256)       590080

 block3_pool (MaxPooling2D)  (None, 22, 22, 256)       0

 block4_conv1 (Conv2D)       (None, 22, 22, 512)       1180160

 block4_conv2 (Conv2D)       (None, 22, 22, 512)       2359808

 block4_conv3 (Conv2D)       (None, 22, 22, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 11, 11, 512)       0

 block5_conv1 (Conv2D)       (None, 11, 11, 512)       2359808

 block5_conv2 (Conv2D)       (None, 11, 11, 512)       2359808

 block5_conv3 (Conv2D)       (None, 11, 11, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 5, 5, 512)         0

=================================================================
Total params: 14714688 (56.13 MB)
Trainable params: 14714688 (56.13 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Pretrained model for feature extraction without data augmentation

In [26]:

```
def get_features_and_labels(dataset):
    all_feature = []
    all_label = []
    for images, labels in dataset:
        preprocessed_images = keras.applications.vgg16.preprocess_input(images)
        features = convoluted_b.predict(preprocessed_images)
        all_feature.append(features)
        all_label.append(labels)
    return np.concatenate(all_feature), np.concatenate(all_label)

train_features, train_labels = get_features_and_labels(train_data)
val_features, val_labels = get_features_and_labels(valid_data)
test_features, test_labels = get_features_and_labels(test_data)
```

```
1/1 [==============================] - 5s 5s/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 34ms/step
```

```
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 3s 3s/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 26ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 2s 2s/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 45ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 31ms/step
```

```
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 18ms/step
```

In [27]:

```
train_features.shape
```

Out[27]:

```
(2000, 5, 5, 512)
```

Model Fitting

In [31]:

```
input_6000 = keras.Input(shape=(5, 5, 512))

d_6000 = layers.Flatten()(input_6000)
d_6000 = layers.Dense(256)(d_6000)
d_6000 = layers.Dropout(0.5)(d_6000)

output_6000 = layers.Dense(1, activation="sigmoid")(d_6000)

model_6000 = keras.Model(input_6000, output_6000)
model_6000.compile(loss="binary_crossentropy",optimizer="rmsprop",metrics=["accurac

callback_6000 = [keras.callbacks.ModelCheckpoint(filepath="feature_extraction.keras

history_6000 = model_6000.fit(train_features, train_labels,epochs=100,validation_da
Epoch 1/100
63/63 [==============================] - 2s 16ms/step - loss: 15.8942 - accuracy:
0.9260 - val_loss: 3.8155 - val_accuracy: 0.9750
Epoch 2/100
63/63 [==============================] - 1s 10ms/step - loss: 2.9894 - accuracy:
0.9765 - val_loss: 3.2367 - val_accuracy: 0.9800
Epoch 3/100
63/63 [==============================] - 0s 8ms/step - loss: 2.5509 - accuracy: 0.
9820 - val_loss: 5.7046 - val_accuracy: 0.9670
Epoch 4/100
63/63 [==============================] - 1s 8ms/step - loss: 1.3793 - accuracy: 0.
9895 - val_loss: 5.1702 - val_accuracy: 0.9730
Epoch 5/100
63/63 [==============================] - 0s 7ms/step - loss: 1.9512 - accuracy: 0.
9875 - val_loss: 5.2021 - val_accuracy: 0.9740
Epoch 6/100
63/63 [==============================] - 0s 8ms/step - loss: 0.3769 - accuracy: 0.
9965 - val_loss: 4.2073 - val_accuracy: 0.9760
Epoch 7/100
63/63 [==============================] - 0s 8ms/step - loss: 0.1154 - accuracy: 0.
9980 - val_loss: 5.0389 - val_accuracy: 0.9750
Epoch 8/100
63/63 [==============================] - 0s 5ms/step - loss: 0.2843 - accuracy: 0.
9970 - val_loss: 4.4403 - val_accuracy: 0.9760
Epoch 9/100
63/63 [==============================] - 0s 5ms/step - loss: 0.1969 - accuracy: 0.
9980 - val_loss: 11.1778 - val_accuracy: 0.9550
Epoch 10/100
63/63 [==============================] - 0s 5ms/step - loss: 0.2609 - accuracy: 0.
9960 - val_loss: 5.8759 - val_accuracy: 0.9730
Epoch 11/100
63/63 [==============================] - 0s 5ms/step - loss: 0.4700 - accuracy: 0.
9950 - val_loss: 5.3212 - val_accuracy: 0.9760
Epoch 12/100
63/63 [==============================] - 0s 6ms/step - loss: 0.4138 - accuracy: 0.
9960 - val_loss: 5.9130 - val_accuracy: 0.9750
Epoch 13/100
63/63 [==============================] - 0s 5ms/step - loss: 0.7518 - accuracy: 0.
9955 - val_loss: 6.0320 - val_accuracy: 0.9730
Epoch 14/100
63/63 [==============================] - 0s 5ms/step - loss: 0.2150 - accuracy: 0.
9980 - val_loss: 5.2540 - val_accuracy: 0.9730
Epoch 15/100
63/63 [==============================] - 0s 5ms/step - loss: 0.3879 - accuracy: 0.
9985 - val_loss: 4.2982 - val_accuracy: 0.9800
Epoch 16/100
63/63 [==============================] - 0s 6ms/step - loss: 0.1497 - accuracy: 0.
9975 - val_loss: 6.0060 - val_accuracy: 0.9760
Epoch 17/100
63/63 [==============================] - 0s 5ms/step - loss: 0.2448 - accuracy: 0.
9980 - val_loss: 5.4009 - val_accuracy: 0.9770
Epoch 18/100
63/63 [==============================] - 0s 6ms/step - loss: 5.4642e-06 - accurac
y: 1.0000 - val_loss: 5.6131 - val_accuracy: 0.9770
Epoch 19/100
63/63 [==============================] - 0s 7ms/step - loss: 0.1525 - accuracy: 0.
```

```
9975 - val_loss: 7.7990 - val_accuracy: 0.9700
Epoch 20/100
63/63 [==============================] - 0s 5ms/step - loss: 0.1800 - accuracy: 0.
9990 - val_loss: 6.0413 - val_accuracy: 0.9720
Epoch 21/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0394 - accuracy: 0.
9990 - val_loss: 6.1265 - val_accuracy: 0.9720
Epoch 22/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0572 - accuracy: 0.
9990 - val_loss: 6.6440 - val_accuracy: 0.9740
Epoch 23/100
63/63 [==============================] - 0s 6ms/step - loss: 0.2705 - accuracy: 0.
9980 - val_loss: 6.4892 - val_accuracy: 0.9740
Epoch 24/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0313 - accuracy: 0.
9995 - val_loss: 5.9434 - val_accuracy: 0.9740
Epoch 25/100
63/63 [==============================] - 0s 5ms/step - loss: 0.2993 - accuracy: 0.
9990 - val_loss: 7.3311 - val_accuracy: 0.9670
Epoch 26/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0152 - accuracy: 0.
9995 - val_loss: 5.4617 - val_accuracy: 0.9760
Epoch 27/100
63/63 [==============================] - 0s 6ms/step - loss: 3.6065e-35 - accurac
y: 1.0000 - val_loss: 5.4617 - val_accuracy: 0.9760
Epoch 28/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0809 - accuracy: 0.
9995 - val_loss: 4.8535 - val_accuracy: 0.9810
Epoch 29/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 4.8535 - val_accuracy: 0.9810
Epoch 30/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0786 - accuracy: 0.
9980 - val_loss: 5.4026 - val_accuracy: 0.9760
Epoch 31/100
63/63 [==============================] - 0s 5ms/step - loss: 1.0967e-18 - accurac
y: 1.0000 - val_loss: 5.4026 - val_accuracy: 0.9760
Epoch 32/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4026 - val_accuracy: 0.9760
Epoch 33/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4026 - val_accuracy: 0.9760
Epoch 34/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4026 - val_accuracy: 0.9760
Epoch 35/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0079 - accuracy: 0.
9995 - val_loss: 6.5020 - val_accuracy: 0.9760
Epoch 36/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0082 - accuracy: 0.
9995 - val_loss: 6.0713 - val_accuracy: 0.9750
Epoch 37/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 6.0713 - val_accuracy: 0.9750
Epoch 38/100
```

```
63/63 [==============================] - 0s 8ms/step - loss: 6.6027e-12 - accurac
y: 1.0000 - val_loss: 6.0713 - val_accuracy: 0.9750
Epoch 39/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0203 - accuracy: 0.
9995 - val_loss: 5.1790 - val_accuracy: 0.9780
Epoch 40/100
63/63 [==============================] - 1s 9ms/step - loss: 4.2692e-28 - accurac
y: 1.0000 - val_loss: 5.1790 - val_accuracy: 0.9780
Epoch 41/100
63/63 [==============================] - 0s 8ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.1790 - val_accuracy: 0.9780
Epoch 42/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.1790 - val_accuracy: 0.9780
Epoch 43/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.1790 - val_accuracy: 0.9780
Epoch 44/100
63/63 [==============================] - 1s 8ms/step - loss: 3.5381e-35 - accurac
y: 1.0000 - val_loss: 5.1790 - val_accuracy: 0.9780
Epoch 45/100
63/63 [==============================] - 0s 8ms/step - loss: 7.2945e-26 - accurac
y: 1.0000 - val_loss: 5.1790 - val_accuracy: 0.9780
Epoch 46/100
63/63 [==============================] - 1s 9ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.1790 - val_accuracy: 0.9780
Epoch 47/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0014 - accuracy: 0.
9995 - val_loss: 5.4653 - val_accuracy: 0.9740
Epoch 48/100
63/63 [==============================] - 0s 8ms/step - loss: 2.3209e-31 - accurac
y: 1.0000 - val_loss: 5.4653 - val_accuracy: 0.9740
Epoch 49/100
63/63 [==============================] - 0s 8ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4653 - val_accuracy: 0.9740
Epoch 50/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0425 - accuracy: 0.
9995 - val_loss: 6.2946 - val_accuracy: 0.9780
Epoch 51/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0598 - accuracy: 0.
9985 - val_loss: 4.7983 - val_accuracy: 0.9750
Epoch 52/100
63/63 [==============================] - 0s 5ms/step - loss: 1.3363e-31 - accurac
y: 1.0000 - val_loss: 4.7983 - val_accuracy: 0.9750
Epoch 53/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 4.7983 - val_accuracy: 0.9750
Epoch 54/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 4.7983 - val_accuracy: 0.9750
Epoch 55/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 4.7983 - val_accuracy: 0.9750
Epoch 56/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 4.7983 - val_accuracy: 0.9750
```

```
Epoch 57/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 4.7983 - val_accuracy: 0.9750
Epoch 58/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 4.7983 - val_accuracy: 0.9750
Epoch 59/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 4.7983 - val_accuracy: 0.9750
Epoch 60/100
63/63 [==============================] - 0s 6ms/step - loss: 8.7853e-17 - accurac
y: 1.0000 - val_loss: 4.7983 - val_accuracy: 0.9750
Epoch 61/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 4.7983 - val_accuracy: 0.9750
Epoch 62/100
63/63 [==============================] - 0s 6ms/step - loss: 0.2182 - accuracy: 0.
9990 - val_loss: 4.7494 - val_accuracy: 0.9750
Epoch 63/100
63/63 [==============================] - 0s 6ms/step - loss: 0.1881 - accuracy: 0.
9990 - val_loss: 5.5386 - val_accuracy: 0.9730
Epoch 64/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.5386 - val_accuracy: 0.9730
Epoch 65/100
63/63 [==============================] - 0s 6ms/step - loss: 1.4973e-25 - accurac
y: 1.0000 - val_loss: 5.5386 - val_accuracy: 0.9730
Epoch 66/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.5386 - val_accuracy: 0.9730
Epoch 67/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.5386 - val_accuracy: 0.9730
Epoch 68/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.5386 - val_accuracy: 0.9730
Epoch 69/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0240 - accuracy: 0.
9995 - val_loss: 5.8546 - val_accuracy: 0.9770
Epoch 70/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0182 - accuracy: 0.
9995 - val_loss: 5.2743 - val_accuracy: 0.9740
Epoch 71/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.2743 - val_accuracy: 0.9740
Epoch 72/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.2743 - val_accuracy: 0.9740
Epoch 73/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.2743 - val_accuracy: 0.9740
Epoch 74/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0241 - accuracy: 0.
9995 - val_loss: 7.3303 - val_accuracy: 0.9730
Epoch 75/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0433 - accuracy: 0.
```

```
9995 - val_loss: 5.3788 - val_accuracy: 0.9750
Epoch 76/100
63/63 [==============================] - 0s 8ms/step - loss: 1.4862e-15 - accurac
y: 1.0000 - val_loss: 5.3788 - val_accuracy: 0.9750
Epoch 77/100
63/63 [==============================] - 0s 8ms/step - loss: 7.6018e-08 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 78/100
63/63 [==============================] - 0s 7ms/step - loss: 4.1875e-20 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 79/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 80/100
63/63 [==============================] - 0s 8ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 81/100
63/63 [==============================] - 1s 8ms/step - loss: 2.3138e-20 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 82/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 83/100
63/63 [==============================] - 1s 9ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 84/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 85/100
63/63 [==============================] - 1s 9ms/step - loss: 1.4202e-24 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 86/100
63/63 [==============================] - 1s 8ms/step - loss: 1.4628e-12 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 87/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 88/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 89/100
63/63 [==============================] - 1s 9ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 90/100
63/63 [==============================] - 0s 8ms/step - loss: 2.8048e-30 - accurac
y: 1.0000 - val_loss: 5.4158 - val_accuracy: 0.9750
Epoch 91/100
63/63 [==============================] - 0s 5ms/step - loss: 8.2030e-10 - accurac
y: 1.0000 - val_loss: 5.4171 - val_accuracy: 0.9740
Epoch 92/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4171 - val_accuracy: 0.9740
Epoch 93/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accurac
y: 1.0000 - val_loss: 5.4171 - val_accuracy: 0.9740
Epoch 94/100
```

```
63/63 [==============================] - 0s 5ms/step - loss: 2.8969e-16 - accurac
y: 1.0000 - val_loss: 5.4171 - val_accuracy: 0.9740
Epoch 95/100
63/63 [==============================] - 0s 7ms/step - loss: 6.5962e-07 - accurac
y: 1.0000 - val_loss: 5.3814 - val_accuracy: 0.9750
Epoch 96/100
63/63 [==============================] - 0s 5ms/step - loss: 1.7048e-36 - accurac
y: 1.0000 - val_loss: 5.3814 - val_accuracy: 0.9750
Epoch 97/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0908 - accuracy: 0.
9990 - val_loss: 5.5859 - val_accuracy: 0.9770
Epoch 98/100
63/63 [==============================] - 0s 5ms/step - loss: 6.8138e-26 - accurac
y: 1.0000 - val_loss: 5.5859 - val_accuracy: 0.9770
Epoch 99/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0338 - accuracy: 0.
9990 - val_loss: 5.8230 - val_accuracy: 0.9760
Epoch 100/100
63/63 [==============================] - 0s 6ms/step - loss: 6.8908e-11 - accurac
y: 1.0000 - val_loss: 5.8229 - val_accuracy: 0.9760
```

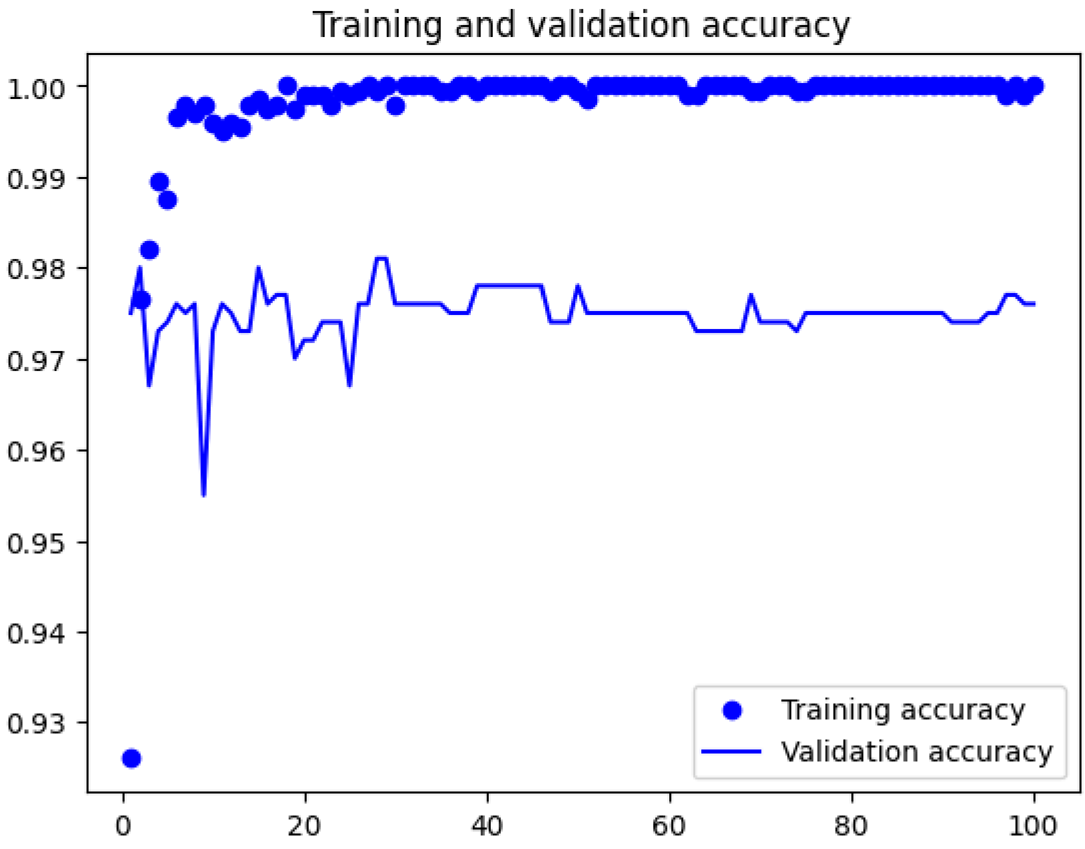
Plot for loss and accuracy during training

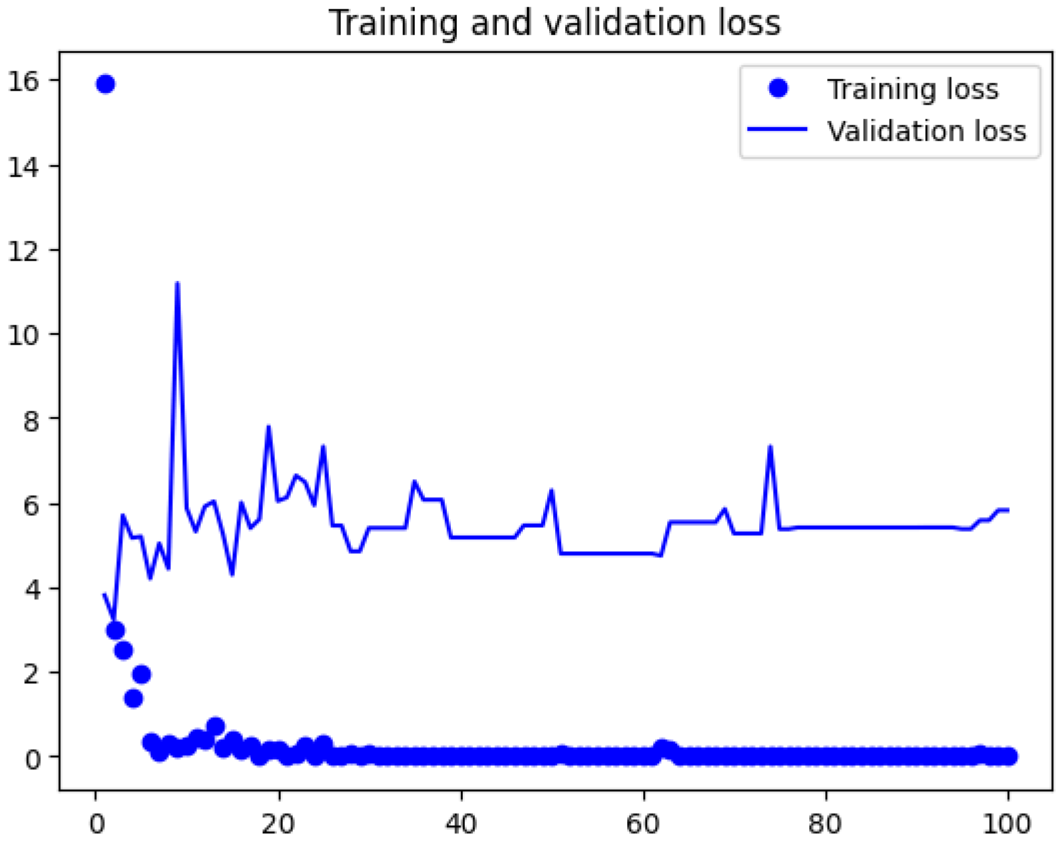
In [33]:

```python
accuracy_6000 = history_6000.history["accuracy"]
valac_6000 = history_6000.history["val_accuracy"]
loss_6000 = history_6000.history["loss"]
valloss_6000 = history_6000.history["val_loss"]

epochs = range(1, len(accuracy_6000) + 1)
plt.plot(epochs, accuracy_6000, "bo", label="Training accuracy")
plt.plot(epochs, valac_6000, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.legend()
plt.figure()

plt.plot(epochs, loss_6000, "bo", label="Training loss")
plt.plot(epochs, valloss_6000, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.legend()
plt.show()
```





Freezing and Unfreezing the Pre-trained Convolutional Base

In [34]:

```
convoluted_base = keras.applications.vgg16.VGG16(weights="imagenet",include_top=Fal

convoluted_base.trainable = False
convoluted_base.trainable = True

print("This is the number of trainable weights ""before freezing the conv base:", l

convoluted_base.trainable = False

print("This is the number of trainable weights ""after freezing the conv base:", le
```

```
This is the number of trainable weights before freezing the conv base: 26
This is the number of trainable weights after freezing the conv base: 0
```

Model is now performing with a classifier and agumentation to convulation base

In [35]:

```
augmented_cb = keras.Sequential([layers.RandomFlip("horizontal"),layers.RandomRotat

input_cb = keras.Input(shape=(180, 180, 3))
d_cb = augmented_cb(input_cb)
d_cb =keras.layers.Lambda(lambda x: keras.applications.vgg16.preprocess_input(x))(d
d_cb = convoluted_base(d_cb)
d_cb = layers.Flatten()(d_cb)
d_cb = layers.Dense(256)(d_cb)
d_cb = layers.Dropout(0.5)(d_cb)

output_cb = layers.Dense(1, activation="sigmoid")(d_cb)

modelx = keras.Model(input_cb, output_cb)

modelx.compile(loss="binary_crossentropy",optimizer="rmsprop",metrics=["accuracy"])

callback_cb = [keras.callbacks.ModelCheckpoint(filepath="features_extraction_with_a

history_cb = modelx.fit(train_data,epochs=100,validation_data=valid_data,callbacks=
Epoch 1/100
63/63 [==============================] - 13s 181ms/step - loss: 15.9558 - accurac
y: 0.8985 - val_loss: 5.3758 - val_accuracy: 0.9590
Epoch 2/100
63/63 [==============================] - 10s 148ms/step - loss: 7.1350 - accuracy:
0.9425 - val_loss: 5.0037 - val_accuracy: 0.9700
Epoch 3/100
63/63 [==============================] - 9s 142ms/step - loss: 5.1418 - accuracy:
0.9545 - val_loss: 3.4774 - val_accuracy: 0.9790
Epoch 4/100
63/63 [==============================] - 9s 145ms/step - loss: 5.8225 - accuracy:
0.9605 - val_loss: 4.1582 - val_accuracy: 0.9720
Epoch 5/100
63/63 [==============================] - 9s 137ms/step - loss: 4.7526 - accuracy:
0.9590 - val_loss: 3.5303 - val_accuracy: 0.9750
Epoch 6/100
63/63 [==============================] - 10s 152ms/step - loss: 3.8000 - accuracy:
0.9655 - val_loss: 3.9895 - val_accuracy: 0.9740
Epoch 7/100
63/63 [==============================] - 9s 139ms/step - loss: 2.9719 - accuracy:
0.9750 - val_loss: 4.0991 - val_accuracy: 0.9740
Epoch 8/100
63/63 [==============================] - 10s 144ms/step - loss: 3.1135 - accuracy:
0.9720 - val_loss: 4.5011 - val_accuracy: 0.9740
Epoch 9/100
63/63 [==============================] - 11s 174ms/step - loss: 1.9564 - accuracy:
0.9790 - val_loss: 4.1722 - val_accuracy: 0.9760
Epoch 10/100
63/63 [==============================] - 9s 141ms/step - loss: 3.0264 - accuracy:
0.9740 - val_loss: 4.9037 - val_accuracy: 0.9700
Epoch 11/100
63/63 [==============================] - 10s 146ms/step - loss: 2.7011 - accuracy:
0.9750 - val_loss: 3.7176 - val_accuracy: 0.9780
Epoch 12/100
63/63 [==============================] - 12s 183ms/step - loss: 1.9212 - accuracy:
0.9780 - val_loss: 3.1333 - val_accuracy: 0.9760
Epoch 13/100
63/63 [==============================] - 10s 154ms/step - loss: 2.9628 - accuracy:
0.9745 - val_loss: 2.8685 - val_accuracy: 0.9800
Epoch 14/100
63/63 [==============================] - 11s 174ms/step - loss: 1.6071 - accuracy:
0.9785 - val_loss: 6.6123 - val_accuracy: 0.9640
Epoch 15/100
63/63 [==============================] - 10s 147ms/step - loss: 1.6899 - accuracy:
0.9795 - val_loss: 3.0412 - val_accuracy: 0.9730
Epoch 16/100
63/63 [==============================] - 10s 148ms/step - loss: 1.3691 - accuracy:
0.9820 - val_loss: 2.8474 - val_accuracy: 0.9730
Epoch 17/100
63/63 [==============================] - 10s 153ms/step - loss: 1.7403 - accuracy:
0.9805 - val_loss: 2.4710 - val_accuracy: 0.9740
Epoch 18/100
63/63 [==============================] - 9s 146ms/step - loss: 1.9358 - accuracy:
0.9800 - val_loss: 3.3686 - val_accuracy: 0.9750
Epoch 19/100
63/63 [==============================] - 10s 150ms/step - loss: 1.6531 - accuracy:
```

```
0.9765 - val_loss: 2.1925 - val_accuracy: 0.9770
Epoch 20/100
63/63 [==============================] - 10s 150ms/step - loss: 1.3840 - accuracy:
0.9795 - val_loss: 2.3737 - val_accuracy: 0.9760
Epoch 21/100
63/63 [==============================] - 9s 144ms/step - loss: 1.1220 - accuracy:
0.9855 - val_loss: 2.6686 - val_accuracy: 0.9790
Epoch 22/100
63/63 [==============================] - 10s 155ms/step - loss: 1.3998 - accuracy:
0.9780 - val_loss: 2.0508 - val_accuracy: 0.9820
Epoch 23/100
63/63 [==============================] - 10s 153ms/step - loss: 1.0577 - accuracy:
0.9850 - val_loss: 1.7928 - val_accuracy: 0.9780
Epoch 24/100
63/63 [==============================] - 10s 146ms/step - loss: 0.8235 - accuracy:
0.9865 - val_loss: 1.9952 - val_accuracy: 0.9760
Epoch 25/100
63/63 [==============================] - 10s 152ms/step - loss: 1.2811 - accuracy:
0.9825 - val_loss: 2.2416 - val_accuracy: 0.9780
Epoch 26/100
63/63 [==============================] - 10s 145ms/step - loss: 1.0674 - accuracy:
0.9805 - val_loss: 2.0345 - val_accuracy: 0.9770
Epoch 27/100
63/63 [==============================] - 10s 153ms/step - loss: 0.8782 - accuracy:
0.9825 - val_loss: 2.1484 - val_accuracy: 0.9810
Epoch 28/100
63/63 [==============================] - 12s 185ms/step - loss: 0.9646 - accuracy:
0.9805 - val_loss: 1.7452 - val_accuracy: 0.9800
Epoch 29/100
63/63 [==============================] - 11s 178ms/step - loss: 0.5824 - accuracy:
0.9885 - val_loss: 2.4584 - val_accuracy: 0.9740
Epoch 30/100
63/63 [==============================] - 10s 147ms/step - loss: 0.7723 - accuracy:
0.9850 - val_loss: 2.0304 - val_accuracy: 0.9800
Epoch 31/100
63/63 [==============================] - 10s 153ms/step - loss: 0.6963 - accuracy:
0.9850 - val_loss: 1.6573 - val_accuracy: 0.9770
Epoch 32/100
63/63 [==============================] - 10s 145ms/step - loss: 0.6035 - accuracy:
0.9880 - val_loss: 1.8722 - val_accuracy: 0.9790
Epoch 33/100
63/63 [==============================] - 10s 152ms/step - loss: 0.5469 - accuracy:
0.9865 - val_loss: 2.8095 - val_accuracy: 0.9740
Epoch 34/100
63/63 [==============================] - 9s 145ms/step - loss: 0.9625 - accuracy:
0.9820 - val_loss: 1.7061 - val_accuracy: 0.9780
Epoch 35/100
63/63 [==============================] - 10s 160ms/step - loss: 0.5439 - accuracy:
0.9855 - val_loss: 1.4380 - val_accuracy: 0.9790
Epoch 36/100
63/63 [==============================] - 11s 177ms/step - loss: 0.4796 - accuracy:
0.9905 - val_loss: 1.9491 - val_accuracy: 0.9780
Epoch 37/100
63/63 [==============================] - 10s 148ms/step - loss: 0.6411 - accuracy:
0.9865 - val_loss: 1.9694 - val_accuracy: 0.9790
Epoch 38/100
```

```
63/63 [==============================] - 11s 179ms/step - loss: 0.6537 - accuracy:
0.9860 - val_loss: 1.6495 - val_accuracy: 0.9780
Epoch 39/100
63/63 [==============================] - 9s 145ms/step - loss: 0.3874 - accuracy:
0.9880 - val_loss: 2.3736 - val_accuracy: 0.9750
Epoch 40/100
63/63 [==============================] - 10s 148ms/step - loss: 0.5110 - accuracy:
0.9925 - val_loss: 1.4638 - val_accuracy: 0.9770
Epoch 41/100
63/63 [==============================] - 10s 150ms/step - loss: 0.3291 - accuracy:
0.9875 - val_loss: 3.6724 - val_accuracy: 0.9670
Epoch 42/100
63/63 [==============================] - 10s 147ms/step - loss: 0.5493 - accuracy:
0.9905 - val_loss: 1.7576 - val_accuracy: 0.9770
Epoch 43/100
63/63 [==============================] - 10s 153ms/step - loss: 0.6936 - accuracy:
0.9860 - val_loss: 1.7450 - val_accuracy: 0.9800
Epoch 44/100
63/63 [==============================] - 11s 178ms/step - loss: 0.4607 - accuracy:
0.9900 - val_loss: 2.1297 - val_accuracy: 0.9750
Epoch 45/100
63/63 [==============================] - 10s 148ms/step - loss: 0.5078 - accuracy:
0.9885 - val_loss: 1.7343 - val_accuracy: 0.9780
Epoch 46/100
63/63 [==============================] - 10s 152ms/step - loss: 0.2475 - accuracy:
0.9900 - val_loss: 2.2320 - val_accuracy: 0.9760
Epoch 47/100
63/63 [==============================] - 10s 147ms/step - loss: 0.8005 - accuracy:
0.9875 - val_loss: 1.6458 - val_accuracy: 0.9770
Epoch 48/100
63/63 [==============================] - 10s 154ms/step - loss: 0.4233 - accuracy:
0.9890 - val_loss: 1.4935 - val_accuracy: 0.9810
Epoch 49/100
63/63 [==============================] - 11s 178ms/step - loss: 0.3908 - accuracy:
0.9905 - val_loss: 1.8785 - val_accuracy: 0.9780
Epoch 50/100
63/63 [==============================] - 10s 148ms/step - loss: 0.6400 - accuracy:
0.9880 - val_loss: 2.0590 - val_accuracy: 0.9760
Epoch 51/100
63/63 [==============================] - 12s 181ms/step - loss: 0.3376 - accuracy:
0.9910 - val_loss: 1.8570 - val_accuracy: 0.9780
Epoch 52/100
63/63 [==============================] - 11s 178ms/step - loss: 0.5319 - accuracy:
0.9870 - val_loss: 1.5247 - val_accuracy: 0.9760
Epoch 53/100
63/63 [==============================] - 11s 179ms/step - loss: 0.3859 - accuracy:
0.9905 - val_loss: 2.2273 - val_accuracy: 0.9790
Epoch 54/100
63/63 [==============================] - 10s 148ms/step - loss: 0.5024 - accuracy:
0.9920 - val_loss: 1.5564 - val_accuracy: 0.9800
Epoch 55/100
63/63 [==============================] - 11s 177ms/step - loss: 0.5192 - accuracy:
0.9885 - val_loss: 1.4583 - val_accuracy: 0.9790
Epoch 56/100
63/63 [==============================] - 10s 145ms/step - loss: 0.6657 - accuracy:
0.9855 - val_loss: 1.6539 - val_accuracy: 0.9750
```

```
Epoch 57/100
63/63 [==============================] - 10s 154ms/step - loss: 0.4522 - accuracy:
0.9915 - val_loss: 1.9946 - val_accuracy: 0.9780
Epoch 58/100
63/63 [==============================] - 11s 177ms/step - loss: 0.3745 - accuracy:
0.9910 - val_loss: 1.9865 - val_accuracy: 0.9790
Epoch 59/100
63/63 [==============================] - 10s 148ms/step - loss: 0.3247 - accuracy:
0.9905 - val_loss: 1.6830 - val_accuracy: 0.9820
Epoch 60/100
63/63 [==============================] - 11s 179ms/step - loss: 0.5247 - accuracy:
0.9880 - val_loss: 1.7878 - val_accuracy: 0.9750
Epoch 61/100
63/63 [==============================] - 10s 149ms/step - loss: 0.1741 - accuracy:
0.9950 - val_loss: 1.8641 - val_accuracy: 0.9770
Epoch 62/100
63/63 [==============================] - 10s 152ms/step - loss: 0.5739 - accuracy:
0.9870 - val_loss: 1.3512 - val_accuracy: 0.9800
Epoch 63/100
63/63 [==============================] - 10s 147ms/step - loss: 0.3265 - accuracy:
0.9945 - val_loss: 1.4327 - val_accuracy: 0.9780
Epoch 64/100
63/63 [==============================] - 11s 179ms/step - loss: 0.4166 - accuracy:
0.9875 - val_loss: 1.9027 - val_accuracy: 0.9750
Epoch 65/100
63/63 [==============================] - 10s 149ms/step - loss: 0.6087 - accuracy:
0.9890 - val_loss: 1.6914 - val_accuracy: 0.9770
Epoch 66/100
63/63 [==============================] - 11s 176ms/step - loss: 0.5223 - accuracy:
0.9865 - val_loss: 1.6376 - val_accuracy: 0.9810
Epoch 67/100
63/63 [==============================] - 10s 145ms/step - loss: 0.2034 - accuracy:
0.9945 - val_loss: 1.4423 - val_accuracy: 0.9800
Epoch 68/100
63/63 [==============================] - 10s 151ms/step - loss: 0.4321 - accuracy:
0.9880 - val_loss: 1.9366 - val_accuracy: 0.9810
Epoch 69/100
63/63 [==============================] - 11s 177ms/step - loss: 0.1751 - accuracy:
0.9950 - val_loss: 2.0963 - val_accuracy: 0.9760
Epoch 70/100
63/63 [==============================] - 10s 151ms/step - loss: 0.3352 - accuracy:
0.9920 - val_loss: 2.2069 - val_accuracy: 0.9780
Epoch 71/100
63/63 [==============================] - 9s 144ms/step - loss: 0.2597 - accuracy:
0.9930 - val_loss: 1.9880 - val_accuracy: 0.9790
Epoch 72/100
63/63 [==============================] - 10s 153ms/step - loss: 0.3330 - accuracy:
0.9950 - val_loss: 2.0613 - val_accuracy: 0.9750
Epoch 73/100
63/63 [==============================] - 11s 177ms/step - loss: 0.1482 - accuracy:
0.9965 - val_loss: 2.8208 - val_accuracy: 0.9760
Epoch 74/100
63/63 [==============================] - 10s 151ms/step - loss: 0.4423 - accuracy:
0.9885 - val_loss: 2.3784 - val_accuracy: 0.9750
Epoch 75/100
63/63 [==============================] - 9s 143ms/step - loss: 0.3767 - accuracy:
```

```
0.9905 - val_loss: 2.0763 - val_accuracy: 0.9770
Epoch 76/100
63/63 [==============================] - 10s 153ms/step - loss: 0.3426 - accuracy:
0.9925 - val_loss: 2.0356 - val_accuracy: 0.9790
Epoch 77/100
63/63 [==============================] - 9s 145ms/step - loss: 0.3502 - accuracy:
0.9910 - val_loss: 2.3754 - val_accuracy: 0.9770
Epoch 78/100
63/63 [==============================] - 10s 148ms/step - loss: 0.2796 - accuracy:
0.9955 - val_loss: 2.3709 - val_accuracy: 0.9770
Epoch 79/100
63/63 [==============================] - 12s 186ms/step - loss: 0.3204 - accuracy:
0.9920 - val_loss: 3.0957 - val_accuracy: 0.9730
Epoch 80/100
63/63 [==============================] - 12s 189ms/step - loss: 0.4406 - accuracy:
0.9895 - val_loss: 2.3595 - val_accuracy: 0.9770
Epoch 81/100
63/63 [==============================] - 11s 179ms/step - loss: 0.3808 - accuracy:
0.9905 - val_loss: 2.5971 - val_accuracy: 0.9780
Epoch 82/100
63/63 [==============================] - 12s 188ms/step - loss: 0.3338 - accuracy:
0.9940 - val_loss: 2.1176 - val_accuracy: 0.9790
Epoch 83/100
63/63 [==============================] - 11s 173ms/step - loss: 0.4850 - accuracy:
0.9905 - val_loss: 2.9414 - val_accuracy: 0.9710
Epoch 84/100
63/63 [==============================] - 10s 161ms/step - loss: 0.5317 - accuracy:
0.9900 - val_loss: 2.2370 - val_accuracy: 0.9800
Epoch 85/100
63/63 [==============================] - 12s 184ms/step - loss: 0.3872 - accuracy:
0.9890 - val_loss: 2.0523 - val_accuracy: 0.9790
Epoch 86/100
63/63 [==============================] - 10s 157ms/step - loss: 0.2865 - accuracy:
0.9925 - val_loss: 2.0579 - val_accuracy: 0.9770
Epoch 87/100
63/63 [==============================] - 10s 148ms/step - loss: 0.4523 - accuracy:
0.9895 - val_loss: 2.2800 - val_accuracy: 0.9760
Epoch 88/100
63/63 [==============================] - 10s 153ms/step - loss: 0.2990 - accuracy:
0.9895 - val_loss: 2.5956 - val_accuracy: 0.9800
Epoch 89/100
63/63 [==============================] - 11s 178ms/step - loss: 0.1916 - accuracy:
0.9945 - val_loss: 2.5974 - val_accuracy: 0.9780
Epoch 90/100
63/63 [==============================] - 10s 150ms/step - loss: 0.4950 - accuracy:
0.9875 - val_loss: 2.6070 - val_accuracy: 0.9770
Epoch 91/100
63/63 [==============================] - 10s 151ms/step - loss: 0.2892 - accuracy:
0.9945 - val_loss: 2.4229 - val_accuracy: 0.9800
Epoch 92/100
63/63 [==============================] - 11s 171ms/step - loss: 0.3135 - accuracy:
0.9930 - val_loss: 2.5974 - val_accuracy: 0.9790
Epoch 93/100
63/63 [==============================] - 10s 152ms/step - loss: 0.2406 - accuracy:
0.9935 - val_loss: 2.3395 - val_accuracy: 0.9800
Epoch 94/100
```

```
63/63 [==============================] - 13s 198ms/step - loss: 0.3796 - accuracy:
0.9930 - val_loss: 2.2084 - val_accuracy: 0.9790
Epoch 95/100
63/63 [==============================] - 12s 184ms/step - loss: 0.3616 - accuracy:
0.9930 - val_loss: 2.8089 - val_accuracy: 0.9770
Epoch 96/100
63/63 [==============================] - 10s 155ms/step - loss: 0.3056 - accuracy:
0.9920 - val_loss: 2.7208 - val_accuracy: 0.9760
Epoch 97/100
63/63 [==============================] - 12s 183ms/step - loss: 0.5678 - accuracy:
0.9890 - val_loss: 3.0098 - val_accuracy: 0.9760
Epoch 98/100
63/63 [==============================] - 10s 153ms/step - loss: 0.1014 - accuracy:
0.9965 - val_loss: 2.4894 - val_accuracy: 0.9820
Epoch 99/100
63/63 [==============================] - 11s 177ms/step - loss: 0.4943 - accuracy:
0.9935 - val_loss: 2.3392 - val_accuracy: 0.9790
Epoch 100/100
63/63 [==============================] - 10s 158ms/step - loss: 0.3284 - accuracy:
0.9940 - val_loss: 2.4799 - val_accuracy: 0.9800
```

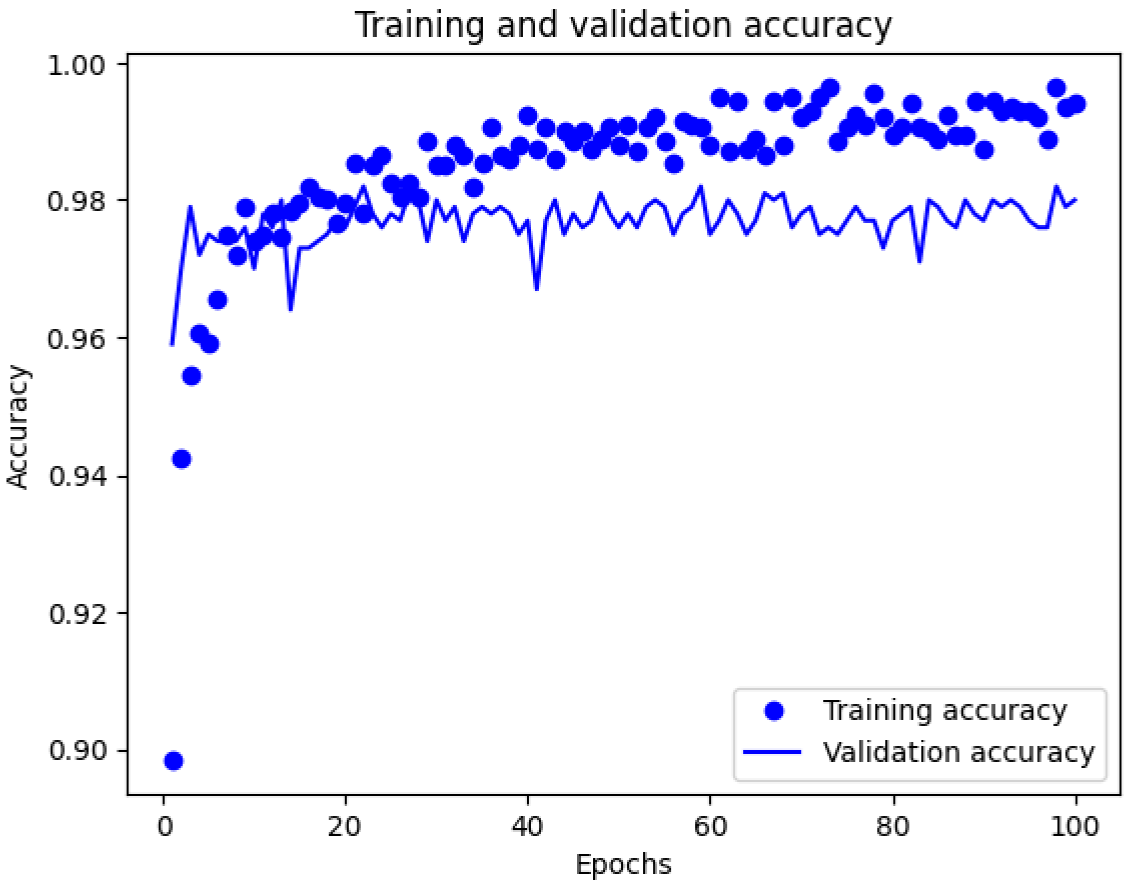
Plot for loss and accuracy during training

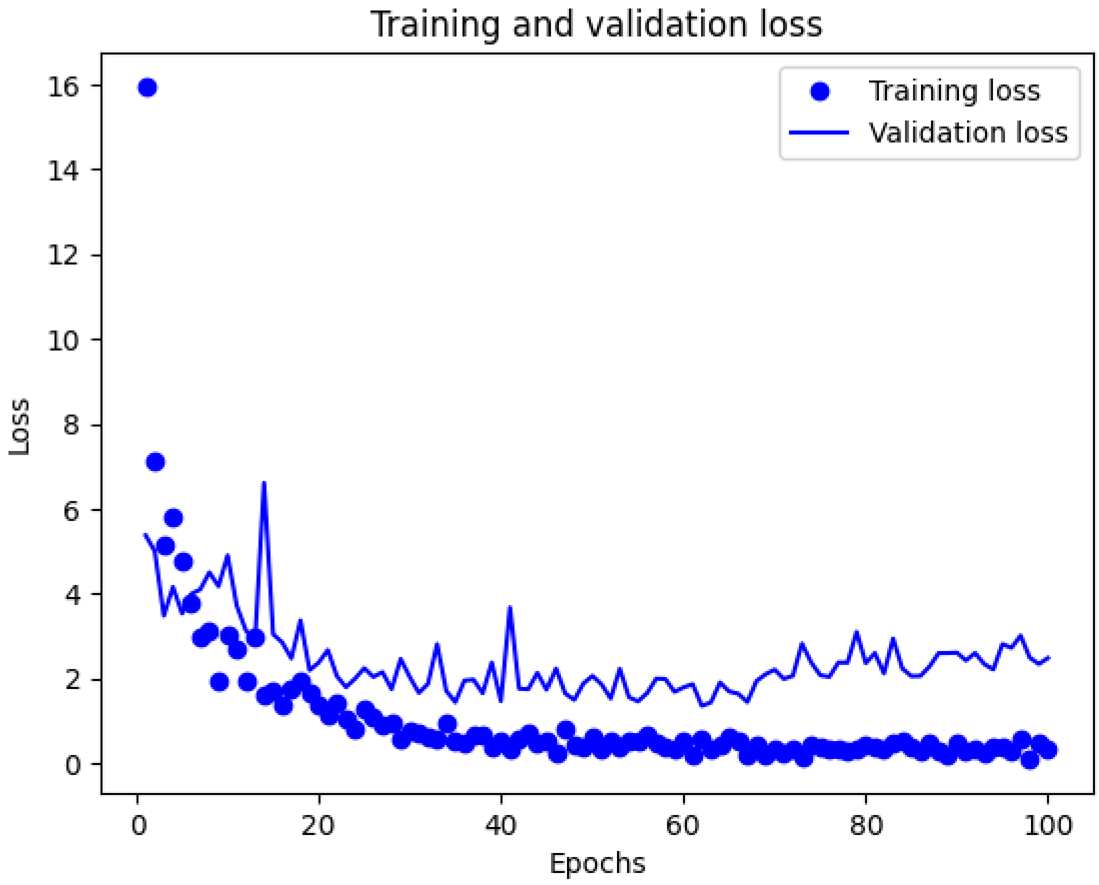
In [36]:

```
accuracy_cb = history_cb.history["accuracy"]
valac_cb = history_cb.history["val_accuracy"]
loss_cb = history_cb.history["loss"]
valloss_cb = history_cb.history["val_loss"]
epochs = range(1, len(accuracy_cb) + 1)

plt.plot(epochs, accuracy_cb, "bo", label="Training accuracy")
plt.plot(epochs, valac_cb, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()

plt.figure()
plt.plot(epochs, loss_cb, "bo", label="Training loss")
plt.plot(epochs, valloss_cb, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```


Training and validation accuracy
1.00
0.98
0.96
0.94
0.92
0.90
Accuracy
0
20
40
60
80
100
Epochs
Training accuracy
Validation accuracy


Training and validation loss
16
14
12
10
8
6
4
2
0
Loss
0
20
40
60
80
100
Epochs
Training loss
Validation loss

Test Accuracy of the model

In [37]:

```
test_cb = keras.models.load_model("features_extraction_with_augmentation2.keras",sa

test_loss, test_acc = test_cb.evaluate(test_data)

print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [==============================] - 4s 86ms/step - loss: 2.1634 - accuracy:
0.9780
Test accuracy: 0.978
```

Fine-tuning a pretrained model

In [38]:

```
convoluted_base.trainable = True

for layer in convoluted_base.layers[:-4]:
    layer.trainable = False

modelx.compile(loss="binary_crossentropy",optimizer=keras.optimizers.RMSprop(learni

callback_tuning = [keras.callbacks.ModelCheckpoint(filepath="fine_tuning.keras",sav

history_tuning = modelx.fit(train_data,epochs=100,validation_data=valid_data,callba
Epoch 1/100
63/63 [==============================] - 16s 203ms/step - loss: 0.5549 - accuracy:
0.9905 - val_loss: 2.3433 - val_accuracy: 0.9790
Epoch 2/100
63/63 [==============================] - 10s 151ms/step - loss: 0.5206 - accuracy:
0.9925 - val_loss: 2.7502 - val_accuracy: 0.9760
Epoch 3/100
63/63 [==============================] - 10s 162ms/step - loss: 0.5802 - accuracy:
0.9910 - val_loss: 2.6274 - val_accuracy: 0.9770
Epoch 4/100
63/63 [==============================] - 10s 151ms/step - loss: 0.1911 - accuracy:
0.9965 - val_loss: 2.5302 - val_accuracy: 0.9760
Epoch 5/100
63/63 [==============================] - 10s 158ms/step - loss: 0.4516 - accuracy:
0.9915 - val_loss: 2.7635 - val_accuracy: 0.9770
Epoch 6/100
63/63 [==============================] - 12s 190ms/step - loss: 0.4487 - accuracy:
0.9930 - val_loss: 2.4606 - val_accuracy: 0.9760
Epoch 7/100
63/63 [==============================] - 11s 173ms/step - loss: 0.2695 - accuracy:
0.9930 - val_loss: 2.2953 - val_accuracy: 0.9760
Epoch 8/100
63/63 [==============================] - 10s 155ms/step - loss: 0.3175 - accuracy:
0.9955 - val_loss: 2.8647 - val_accuracy: 0.9800
Epoch 9/100
63/63 [==============================] - 11s 165ms/step - loss: 0.3778 - accuracy:
0.9935 - val_loss: 2.5146 - val_accuracy: 0.9770
Epoch 10/100
63/63 [==============================] - 12s 189ms/step - loss: 0.5305 - accuracy:
0.9920 - val_loss: 2.3863 - val_accuracy: 0.9760
Epoch 11/100
63/63 [==============================] - 11s 163ms/step - loss: 0.2447 - accuracy:
0.9920 - val_loss: 2.5387 - val_accuracy: 0.9770
Epoch 12/100
63/63 [==============================] - 12s 189ms/step - loss: 0.1030 - accuracy:
0.9955 - val_loss: 2.6883 - val_accuracy: 0.9760
Epoch 13/100
63/63 [==============================] - 11s 159ms/step - loss: 0.3354 - accuracy:
0.9930 - val_loss: 2.9982 - val_accuracy: 0.9760
Epoch 14/100
63/63 [==============================] - 12s 187ms/step - loss: 0.1639 - accuracy:
0.9945 - val_loss: 2.8970 - val_accuracy: 0.9760
Epoch 15/100
63/63 [==============================] - 10s 156ms/step - loss: 0.3767 - accuracy:
0.9950 - val_loss: 2.5562 - val_accuracy: 0.9740
Epoch 16/100
63/63 [==============================] - 11s 175ms/step - loss: 0.3794 - accuracy:
0.9925 - val_loss: 2.1771 - val_accuracy: 0.9730
Epoch 17/100
63/63 [==============================] - 11s 167ms/step - loss: 0.1924 - accuracy:
0.9955 - val_loss: 2.1703 - val_accuracy: 0.9780
Epoch 18/100
63/63 [==============================] - 11s 167ms/step - loss: 0.2351 - accuracy:
0.9955 - val_loss: 2.1909 - val_accuracy: 0.9780
Epoch 19/100
63/63 [==============================] - 11s 170ms/step - loss: 0.1733 - accuracy:
```

```
0.9945 - val_loss: 1.9651 - val_accuracy: 0.9790
Epoch 20/100
63/63 [==============================] - 11s 166ms/step - loss: 0.0637 - accuracy:
0.9965 - val_loss: 2.1805 - val_accuracy: 0.9820
Epoch 21/100
63/63 [==============================] - 12s 189ms/step - loss: 0.2340 - accuracy:
0.9950 - val_loss: 2.3585 - val_accuracy: 0.9790
Epoch 22/100
63/63 [==============================] - 11s 160ms/step - loss: 0.1215 - accuracy:
0.9970 - val_loss: 2.0835 - val_accuracy: 0.9780
Epoch 23/100
63/63 [==============================] - 11s 166ms/step - loss: 0.1454 - accuracy:
0.9960 - val_loss: 2.1496 - val_accuracy: 0.9780
Epoch 24/100
63/63 [==============================] - 12s 190ms/step - loss: 0.0561 - accuracy:
0.9970 - val_loss: 2.1183 - val_accuracy: 0.9800
Epoch 25/100
63/63 [==============================] - 11s 161ms/step - loss: 0.1037 - accuracy:
0.9975 - val_loss: 2.2702 - val_accuracy: 0.9810
Epoch 26/100
63/63 [==============================] - 10s 162ms/step - loss: 0.1367 - accuracy:
0.9960 - val_loss: 2.2398 - val_accuracy: 0.9800
Epoch 27/100
63/63 [==============================] - 10s 160ms/step - loss: 0.1864 - accuracy:
0.9945 - val_loss: 2.2315 - val_accuracy: 0.9800
Epoch 28/100
63/63 [==============================] - 11s 169ms/step - loss: 0.1923 - accuracy:
0.9955 - val_loss: 2.0679 - val_accuracy: 0.9830
Epoch 29/100
63/63 [==============================] - 12s 192ms/step - loss: 0.0465 - accuracy:
0.9995 - val_loss: 2.1125 - val_accuracy: 0.9790
Epoch 30/100
63/63 [==============================] - 13s 204ms/step - loss: 0.1554 - accuracy:
0.9970 - val_loss: 1.9145 - val_accuracy: 0.9820
Epoch 31/100
63/63 [==============================] - 11s 173ms/step - loss: 0.1361 - accuracy:
0.9975 - val_loss: 1.9088 - val_accuracy: 0.9820
Epoch 32/100
63/63 [==============================] - 11s 162ms/step - loss: 0.2466 - accuracy:
0.9950 - val_loss: 2.2450 - val_accuracy: 0.9770
Epoch 33/100
63/63 [==============================] - 10s 159ms/step - loss: 0.0140 - accuracy:
0.9975 - val_loss: 2.0648 - val_accuracy: 0.9780
Epoch 34/100
63/63 [==============================] - 11s 161ms/step - loss: 0.1562 - accuracy:
0.9970 - val_loss: 1.9791 - val_accuracy: 0.9800
Epoch 35/100
63/63 [==============================] - 11s 167ms/step - loss: 0.0923 - accuracy:
0.9985 - val_loss: 2.0643 - val_accuracy: 0.9790
Epoch 36/100
63/63 [==============================] - 11s 165ms/step - loss: 0.0304 - accuracy:
0.9975 - val_loss: 2.0832 - val_accuracy: 0.9780
Epoch 37/100
63/63 [==============================] - 10s 155ms/step - loss: 0.0501 - accuracy:
0.9990 - val_loss: 2.1683 - val_accuracy: 0.9790
Epoch 38/100
```

```
63/63 [==============================] - 11s 167ms/step - loss: 0.0573 - accuracy:
0.9970 - val_loss: 2.4853 - val_accuracy: 0.9770
Epoch 39/100
63/63 [==============================] - 12s 190ms/step - loss: 0.0674 - accuracy:
0.9985 - val_loss: 1.9955 - val_accuracy: 0.9800
Epoch 40/100
63/63 [==============================] - 10s 157ms/step - loss: 0.0066 - accuracy:
0.9995 - val_loss: 2.2838 - val_accuracy: 0.9780
Epoch 41/100
63/63 [==============================] - 11s 167ms/step - loss: 0.0826 - accuracy:
0.9980 - val_loss: 1.9633 - val_accuracy: 0.9790
Epoch 42/100
63/63 [==============================] - 12s 192ms/step - loss: 0.1634 - accuracy:
0.9980 - val_loss: 2.0371 - val_accuracy: 0.9810
Epoch 43/100
63/63 [==============================] - 10s 160ms/step - loss: 0.2393 - accuracy:
0.9955 - val_loss: 2.1234 - val_accuracy: 0.9790
Epoch 44/100
63/63 [==============================] - 11s 167ms/step - loss: 0.1141 - accuracy:
0.9965 - val_loss: 2.3200 - val_accuracy: 0.9800
Epoch 45/100
63/63 [==============================] - 11s 164ms/step - loss: 8.4495e-04 - accur
acy: 0.9995 - val_loss: 2.0218 - val_accuracy: 0.9810
Epoch 46/100
63/63 [==============================] - 11s 162ms/step - loss: 0.0396 - accuracy:
0.9990 - val_loss: 2.0388 - val_accuracy: 0.9820
Epoch 47/100
63/63 [==============================] - 11s 169ms/step - loss: 0.1067 - accuracy:
0.9965 - val_loss: 2.0191 - val_accuracy: 0.9810
Epoch 48/100
63/63 [==============================] - 12s 192ms/step - loss: 0.1144 - accuracy:
0.9960 - val_loss: 2.2451 - val_accuracy: 0.9790
Epoch 49/100
63/63 [==============================] - 11s 165ms/step - loss: 0.1251 - accuracy:
0.9980 - val_loss: 2.1640 - val_accuracy: 0.9800
Epoch 50/100
63/63 [==============================] - 12s 190ms/step - loss: 0.0334 - accuracy:
0.9975 - val_loss: 2.0159 - val_accuracy: 0.9800
Epoch 51/100
63/63 [==============================] - 11s 162ms/step - loss: 0.0707 - accuracy:
0.9970 - val_loss: 3.4297 - val_accuracy: 0.9740
Epoch 52/100
63/63 [==============================] - 11s 169ms/step - loss: 0.0239 - accuracy:
0.9980 - val_loss: 2.2787 - val_accuracy: 0.9800
Epoch 53/100
63/63 [==============================] - 12s 191ms/step - loss: 0.2722 - accuracy:
0.9945 - val_loss: 2.1909 - val_accuracy: 0.9800
Epoch 54/100
63/63 [==============================] - 11s 167ms/step - loss: 0.0528 - accuracy:
0.9995 - val_loss: 2.1648 - val_accuracy: 0.9810
Epoch 55/100
63/63 [==============================] - 12s 191ms/step - loss: 0.0425 - accuracy:
0.9975 - val_loss: 2.2019 - val_accuracy: 0.9820
Epoch 56/100
63/63 [==============================] - 11s 163ms/step - loss: 0.0369 - accuracy:
0.9990 - val_loss: 2.3680 - val_accuracy: 0.9820
```

```
Epoch 57/100
63/63 [==============================] - 13s 198ms/step - loss: 0.0519 - accuracy:
0.9990 - val_loss: 2.3461 - val_accuracy: 0.9840
Epoch 58/100
63/63 [==============================] - 11s 165ms/step - loss: 0.0772 - accuracy:
0.9980 - val_loss: 3.0075 - val_accuracy: 0.9770
Epoch 59/100
63/63 [==============================] - 12s 195ms/step - loss: 0.1472 - accuracy:
0.9975 - val_loss: 2.4610 - val_accuracy: 0.9790
Epoch 60/100
63/63 [==============================] - 10s 159ms/step - loss: 0.0479 - accuracy:
0.9990 - val_loss: 2.2722 - val_accuracy: 0.9810
Epoch 61/100
63/63 [==============================] - 11s 171ms/step - loss: 0.0611 - accuracy:
0.9975 - val_loss: 2.1657 - val_accuracy: 0.9800
Epoch 62/100
63/63 [==============================] - 11s 178ms/step - loss: 0.0927 - accuracy:
0.9970 - val_loss: 2.2582 - val_accuracy: 0.9790
Epoch 63/100
63/63 [==============================] - 11s 172ms/step - loss: 0.0192 - accuracy:
0.9975 - val_loss: 2.2362 - val_accuracy: 0.9780
Epoch 64/100
63/63 [==============================] - 11s 162ms/step - loss: 0.0128 - accuracy:
0.9990 - val_loss: 2.1969 - val_accuracy: 0.9790
Epoch 65/100
63/63 [==============================] - 12s 191ms/step - loss: 0.0719 - accuracy:
0.9980 - val_loss: 2.0509 - val_accuracy: 0.9800
Epoch 66/100
63/63 [==============================] - 11s 164ms/step - loss: 0.0264 - accuracy:
0.9985 - val_loss: 2.0938 - val_accuracy: 0.9770
Epoch 67/100
63/63 [==============================] - 13s 207ms/step - loss: 0.1415 - accuracy:
0.9965 - val_loss: 1.8861 - val_accuracy: 0.9810
Epoch 68/100
63/63 [==============================] - 10s 162ms/step - loss: 0.0771 - accuracy:
0.9990 - val_loss: 1.9563 - val_accuracy: 0.9810
Epoch 69/100
63/63 [==============================] - 12s 180ms/step - loss: 0.0486 - accuracy:
0.9985 - val_loss: 2.4428 - val_accuracy: 0.9770
Epoch 70/100
63/63 [==============================] - 12s 189ms/step - loss: 0.0861 - accuracy:
0.9980 - val_loss: 2.0963 - val_accuracy: 0.9800
Epoch 71/100
63/63 [==============================] - 11s 160ms/step - loss: 0.0402 - accuracy:
0.9980 - val_loss: 2.2322 - val_accuracy: 0.9820
Epoch 72/100
63/63 [==============================] - 12s 194ms/step - loss: 0.0272 - accuracy:
0.9990 - val_loss: 2.4352 - val_accuracy: 0.9790
Epoch 73/100
63/63 [==============================] - 12s 190ms/step - loss: 0.0816 - accuracy:
0.9980 - val_loss: 2.3556 - val_accuracy: 0.9790
Epoch 74/100
63/63 [==============================] - 13s 196ms/step - loss: 0.0275 - accuracy:
0.9990 - val_loss: 2.2228 - val_accuracy: 0.9820
Epoch 75/100
63/63 [==============================] - 10s 161ms/step - loss: 0.0445 - accuracy:
```

```
0.9980 - val_loss: 3.5060 - val_accuracy: 0.9730
Epoch 76/100
63/63 [==============================] - 11s 173ms/step - loss: 0.0428 - accuracy:
0.9985 - val_loss: 1.8639 - val_accuracy: 0.9830
Epoch 77/100
63/63 [==============================] - 12s 194ms/step - loss: 0.0202 - accuracy:
0.9995 - val_loss: 2.5495 - val_accuracy: 0.9770
Epoch 78/100
63/63 [==============================] - 10s 158ms/step - loss: 0.0333 - accuracy:
0.9985 - val_loss: 2.1774 - val_accuracy: 0.9800
Epoch 79/100
63/63 [==============================] - 11s 164ms/step - loss: 0.1676 - accuracy:
0.9970 - val_loss: 2.2085 - val_accuracy: 0.9800
Epoch 80/100
63/63 [==============================] - 11s 178ms/step - loss: 0.1140 - accuracy:
0.9975 - val_loss: 1.7882 - val_accuracy: 0.9810
Epoch 81/100
63/63 [==============================] - 10s 159ms/step - loss: 8.0519e-04 - accur
acy: 0.9995 - val_loss: 1.8860 - val_accuracy: 0.9800
Epoch 82/100
63/63 [==============================] - 12s 177ms/step - loss: 0.0220 - accuracy:
0.9985 - val_loss: 2.3576 - val_accuracy: 0.9790
Epoch 83/100
63/63 [==============================] - 13s 194ms/step - loss: 0.0219 - accuracy:
0.9985 - val_loss: 1.8434 - val_accuracy: 0.9770
Epoch 84/100
63/63 [==============================] - 12s 176ms/step - loss: 0.0292 - accuracy:
0.9970 - val_loss: 2.3097 - val_accuracy: 0.9790
Epoch 85/100
63/63 [==============================] - 12s 195ms/step - loss: 0.0181 - accuracy:
0.9995 - val_loss: 1.9521 - val_accuracy: 0.9780
Epoch 86/100
63/63 [==============================] - 10s 160ms/step - loss: 0.0611 - accuracy:
0.9980 - val_loss: 1.8166 - val_accuracy: 0.9800
Epoch 87/100
63/63 [==============================] - 11s 166ms/step - loss: 0.0378 - accuracy:
0.9990 - val_loss: 1.8394 - val_accuracy: 0.9760
Epoch 88/100
63/63 [==============================] - 11s 167ms/step - loss: 0.0327 - accuracy:
0.9990 - val_loss: 1.9085 - val_accuracy: 0.9780
Epoch 89/100
63/63 [==============================] - 12s 192ms/step - loss: 0.0369 - accuracy:
0.9980 - val_loss: 2.1029 - val_accuracy: 0.9770
Epoch 90/100
63/63 [==============================] - 11s 166ms/step - loss: 0.0524 - accuracy:
0.9975 - val_loss: 2.0428 - val_accuracy: 0.9770
Epoch 91/100
63/63 [==============================] - 10s 160ms/step - loss: 0.0259 - accuracy:
0.9990 - val_loss: 2.2238 - val_accuracy: 0.9760
Epoch 92/100
63/63 [==============================] - 11s 167ms/step - loss: 7.5075e-07 - accur
acy: 1.0000 - val_loss: 2.2018 - val_accuracy: 0.9780
Epoch 93/100
63/63 [==============================] - 12s 193ms/step - loss: 8.5655e-05 - accur
acy: 1.0000 - val_loss: 2.2585 - val_accuracy: 0.9800
Epoch 94/100
```

```
63/63 [==============================] - 11s 169ms/step - loss: 0.0410 - accuracy:
0.9990 - val_loss: 2.2444 - val_accuracy: 0.9800
Epoch 95/100
63/63 [==============================] - 12s 191ms/step - loss: 0.1360 - accuracy:
0.9970 - val_loss: 2.2405 - val_accuracy: 0.9790
Epoch 96/100
63/63 [==============================] - 11s 166ms/step - loss: 0.0567 - accuracy:
0.9985 - val_loss: 2.6237 - val_accuracy: 0.9770
Epoch 97/100
63/63 [==============================] - 11s 162ms/step - loss: 0.0284 - accuracy:
0.9990 - val_loss: 2.7267 - val_accuracy: 0.9780
Epoch 98/100
63/63 [==============================] - 10s 156ms/step - loss: 0.0476 - accuracy:
0.9985 - val_loss: 2.4855 - val_accuracy: 0.9750
Epoch 99/100
63/63 [==============================] - 11s 165ms/step - loss: 0.0538 - accuracy:
0.9990 - val_loss: 2.2291 - val_accuracy: 0.9800
Epoch 100/100
63/63 [==============================] - 12s 191ms/step - loss: 0.0911 - accuracy:
0.9970 - val_loss: 2.2229 - val_accuracy: 0.9800
```

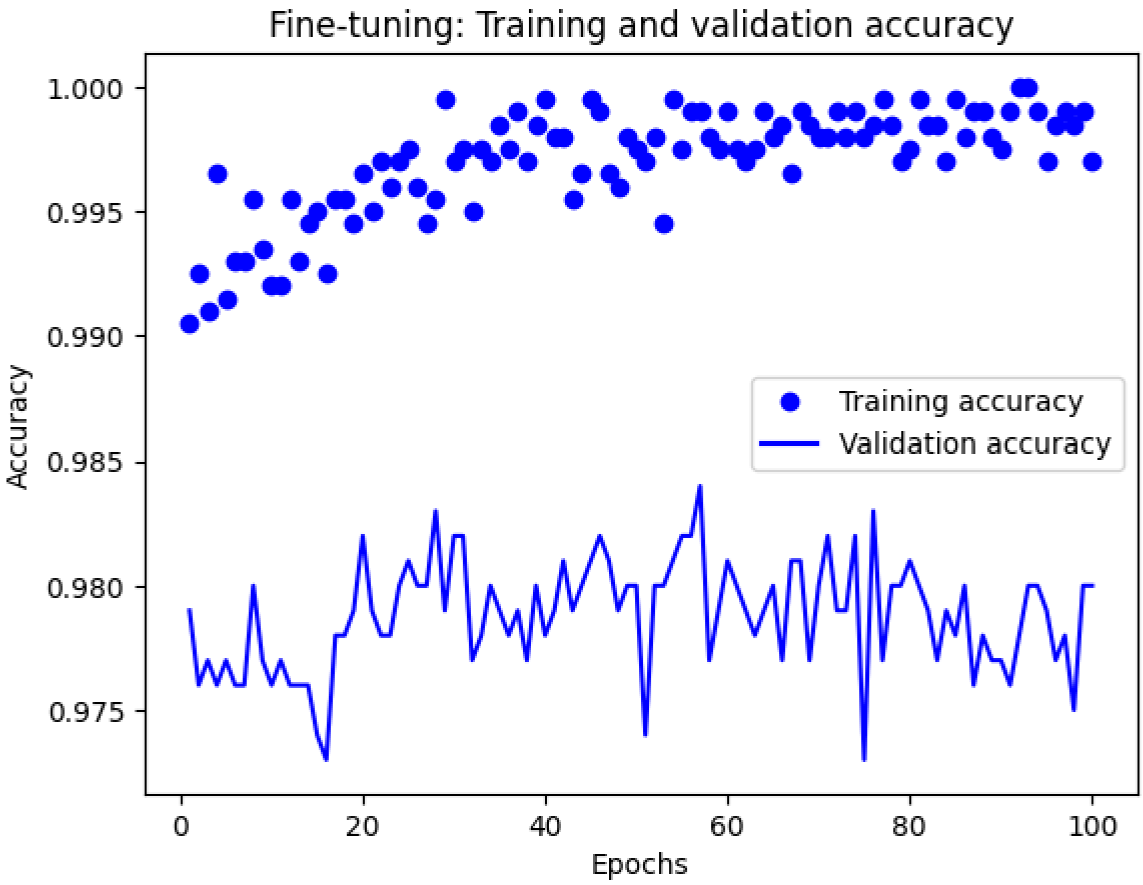
Plot for loss and accuracy during training

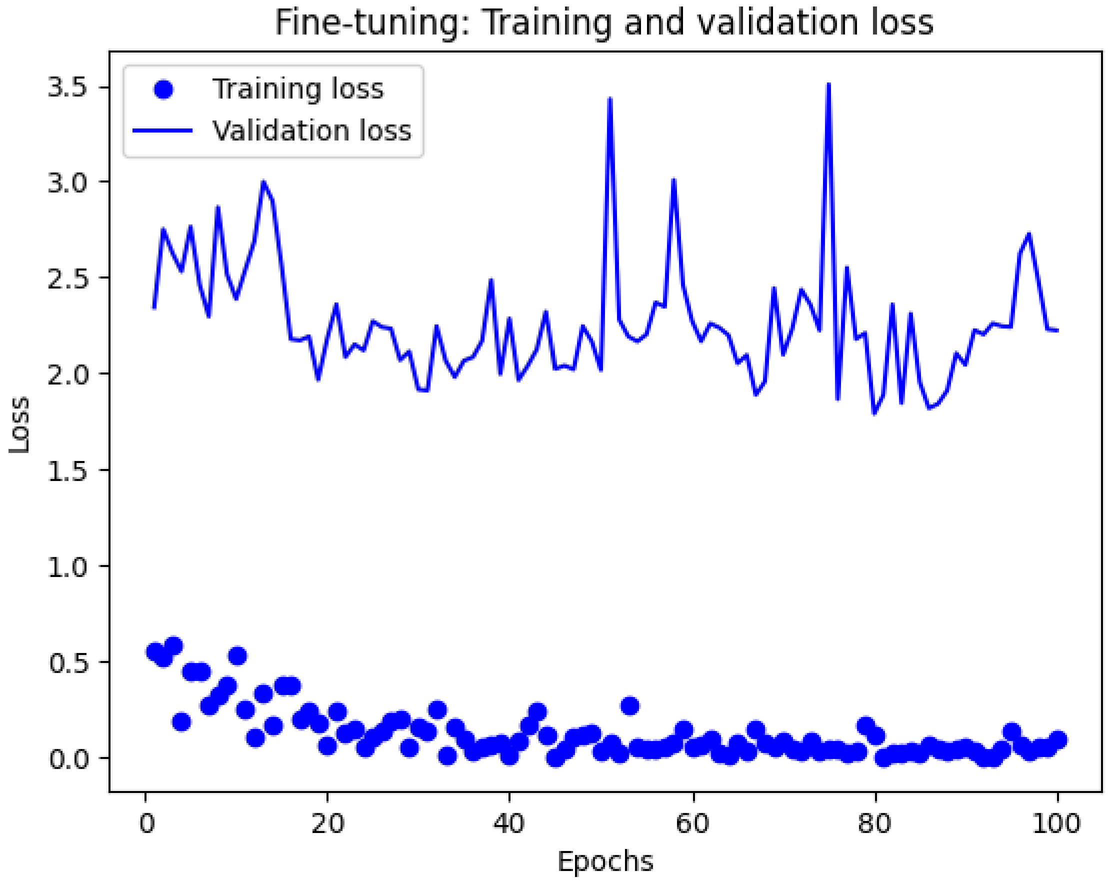
In [39]:

```python
accuracy_tuning = history_tuning.history["accuracy"]
val_tuning = history_tuning.history["val_accuracy"]
loss_tuning = history_tuning.history["loss"]
val_loss_tuning = history_tuning.history["val_loss"]
epochs = range(1, len(accuracy_tuning) + 1)

plt.plot(epochs, accuracy_tuning, "bo", label="Training accuracy")
plt.plot(epochs, val_tuning, "b", label="Validation accuracy")
plt.title("Fine-tuning: Training and validation accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()

plt.figure()
plt.plot(epochs, loss_tuning, "bo", label="Training loss")
plt.plot(epochs, val_loss_tuning, "b", label="Validation loss")
plt.title("Fine-tuning: Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```


Fine-tuning: Training and validation accuracy
Accuracy
1.000
0.995
0.990
0.985
0.980
0.975
0
20
40
60
80
100
Epochs
Training accuracy
Validation accuracy


Fine-tuning: Training and validation loss
Loss
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
Epochs
Training loss
Validation loss

Test Accuracy of model

In [40]:

```
test_tuning = keras.models.load_model("fine_tuning.keras",safe_mode=False)

test_loss, test_acc = test_tuning.evaluate(test_data)

print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [==============================] - 3s 89ms/step - loss: 3.0089 - accuracy:
0.9780
Test accuracy: 0.978
```