Convolution Assignment

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

In [5]:

```
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 (1).json
```

Out[5]:

```
{'kaggle (1).json': b'{"username":"srinagadattugummadi","key":"b8ddb013cb25a6520
145fc825e577858"}'}
```

In [6]:

```
!mkdir ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
```

```
mkdir: cannot create directory ‘/root/.kaggle’: File exists
```

In [7]:

```
!kaggle competitions download -c dogs-vs-cats
!unzip -qq dogs-vs-cats.zip
!unzip -qq train.zip
```

```
Downloading dogs-vs-cats.zip to /content
... resuming from 234881024 bytes (616695665 bytes left) ...
100% 812M/812M [00:27<00:00, 24.8MB/s]
100% 812M/812M [00:27<00:00, 22.8MB/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 [8]:

```
import os, shutil, pathlib
d_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 = d_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 [9]:

```
from tensorflow.keras.utils import image_dataset_from_directory

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

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

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

```
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 [10]:

```
import numpy as np
import tensorflow as tf

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

batch_data = dataset.batch(32)
for i, element in enumerate(batch_data):
    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 [11]:

```
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 [12]:

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

input_1000 = keras.Input(shape=(180, 180, 3))
dat = layers.Rescaling(1./255)(input_1000)
dat = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(dat)
dat = layers.MaxPooling2D(pool_size=2)(dat)
dat = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(dat)
dat = layers.MaxPooling2D(pool_size=2)(dat)
dat = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(dat)
dat = layers.MaxPooling2D(pool_size=2)(dat)
dat = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(dat)
dat = layers.MaxPooling2D(pool_size=2)(dat)
dat = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(dat)
dat = layers.Flatten()(dat)
dat = layers.Dropout(0.5)(dat)
output_1000 = layers.Dense(1, activation="sigmoid")(dat)
model_5555 = keras.Model(inputs=input_1000, outputs=output_1000)
```

Model Training

In [13]:

```
model_5555.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 T4 GPU to reduce the time it takes for each epoch to execute

In [14]:

```
model_5555.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 [15]:

```
callback_5555 = [
keras.callbacks.ModelCheckpoint(
filepath="convnet_from_scratch.keras",
save_best_only=True,
monitor="val_loss")
]
history_5555 = model_5555.fit(train_data,
epochs=100,
validation_data=valid_data,
callbacks=callback_5555)
```

```
Epoch 1/100
63/63 [==============================] - 15s 85ms/step - loss: 0.6918 - accurac
y: 0.5275 - val_loss: 0.6910 - val_accuracy: 0.5420
Epoch 2/100
63/63 [==============================] - 7s 101ms/step - loss: 0.6806 - accurac
y: 0.5680 - val_loss: 0.6966 - val_accuracy: 0.5110
Epoch 3/100
63/63 [==============================] - 5s 68ms/step - loss: 0.6808 - accuracy:
0.5520 - val_loss: 0.6619 - val_accuracy: 0.6060
Epoch 4/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6372 - accuracy:
0.6275 - val_loss: 0.6604 - val_accuracy: 0.5990
Epoch 5/100
63/63 [==============================] - 5s 70ms/step - loss: 0.6074 - accuracy:
0.6760 - val_loss: 0.5982 - val_accuracy: 0.6960
Epoch 6/100
63/63 [==============================] - 6s 89ms/step - loss: 0.5640 - accuracy:
0.7020 - val_loss: 0.6028 - val_accuracy: 0.6880
Epoch 7/100
63/63 [==============================] - 4s 59ms/step - loss: 0.5516 - accuracy:
0.7085 - val_loss: 0.6186 - val_accuracy: 0.6790
Epoch 8/100
63/63 [==============================] - 4s 58ms/step - loss: 0.5323 - accuracy:
0.7405 - val_loss: 0.6069 - val_accuracy: 0.6960
Epoch 9/100
63/63 [==============================] - 6s 98ms/step - loss: 0.4686 - accuracy:
0.7755 - val_loss: 0.6299 - val_accuracy: 0.6950
Epoch 10/100
63/63 [==============================] - 4s 58ms/step - loss: 0.4487 - accuracy:
0.7850 - val_loss: 0.6054 - val_accuracy: 0.7100
```

```
Epoch 11/100
63/63 [==============================] - 6s 88ms/step - loss: 0.3966 - accuracy:
0.8235 - val_loss: 0.6347 - val_accuracy: 0.6960
Epoch 12/100
63/63 [==============================] - 4s 59ms/step - loss: 0.3671 - accuracy:
0.8325 - val_loss: 0.6568 - val_accuracy: 0.7170
Epoch 13/100
63/63 [==============================] - 4s 57ms/step - loss: 0.3245 - accuracy:
0.8490 - val_loss: 0.6815 - val_accuracy: 0.7070
Epoch 14/100
63/63 [==============================] - 7s 108ms/step - loss: 0.2848 - accurac
y: 0.8735 - val_loss: 0.6901 - val_accuracy: 0.7200
Epoch 15/100
63/63 [==============================] - 4s 57ms/step - loss: 0.2538 - accuracy:
0.8915 - val_loss: 0.7532 - val_accuracy: 0.7080
Epoch 16/100
63/63 [==============================] - 6s 86ms/step - loss: 0.1898 - accuracy:
0.9155 - val_loss: 0.9122 - val_accuracy: 0.7060
Epoch 17/100
63/63 [==============================] - 4s 58ms/step - loss: 0.1950 - accuracy:
0.9180 - val_loss: 0.9185 - val_accuracy: 0.7110
Epoch 18/100
63/63 [==============================] - 5s 82ms/step - loss: 0.1324 - accuracy:
0.9465 - val_loss: 0.9169 - val_accuracy: 0.7460
Epoch 19/100
63/63 [==============================] - 6s 85ms/step - loss: 0.1449 - accuracy:
0.9435 - val_loss: 0.9515 - val_accuracy: 0.7100
Epoch 20/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0956 - accuracy:
0.9615 - val_loss: 1.1390 - val_accuracy: 0.7230
Epoch 21/100
63/63 [==============================] - 4s 59ms/step - loss: 0.0830 - accuracy:
0.9695 - val_loss: 1.1413 - val_accuracy: 0.7300
Epoch 22/100
63/63 [==============================] - 7s 105ms/step - loss: 0.1017 - accurac
y: 0.9605 - val_loss: 1.0232 - val_accuracy: 0.7470
Epoch 23/100
63/63 [==============================] - 5s 64ms/step - loss: 0.0753 - accuracy:
0.9715 - val_loss: 1.2045 - val_accuracy: 0.7340
Epoch 24/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0454 - accuracy:
0.9825 - val_loss: 1.2451 - val_accuracy: 0.7370
Epoch 25/100
63/63 [==============================] - 5s 83ms/step - loss: 0.0446 - accuracy:
0.9855 - val_loss: 1.2965 - val_accuracy: 0.7400
Epoch 26/100
63/63 [==============================] - 6s 88ms/step - loss: 0.0538 - accuracy:
0.9790 - val_loss: 1.4615 - val_accuracy: 0.7390
Epoch 27/100
63/63 [==============================] - 4s 61ms/step - loss: 0.0422 - accuracy:
0.9845 - val_loss: 1.4545 - val_accuracy: 0.7270
Epoch 28/100
63/63 [==============================] - 7s 105ms/step - loss: 0.0644 - accurac
y: 0.9810 - val_loss: 1.1820 - val_accuracy: 0.7360
Epoch 29/100
63/63 [==============================] - 5s 64ms/step - loss: 0.0415 - accuracy:
0.9845 - val_loss: 1.3365 - val_accuracy: 0.7380
Epoch 30/100
63/63 [==============================] - 4s 59ms/step - loss: 0.0318 - accuracy:
0.9895 - val_loss: 1.2181 - val_accuracy: 0.7420
Epoch 31/100
63/63 [==============================] - 5s 70ms/step - loss: 0.0200 - accuracy:
0.9955 - val_loss: 1.5184 - val_accuracy: 0.7240
Epoch 32/100
63/63 [==============================] - 6s 93ms/step - loss: 0.0325 - accuracy:
```

```
0.9900 - val_loss: 1.3818 - val_accuracy: 0.7290
Epoch 33/100
63/63 [==============================] - 4s 60ms/step - loss: 0.0400 - accuracy:
0.9850 - val_loss: 1.7396 - val_accuracy: 0.7140
Epoch 34/100
63/63 [==============================] - 6s 95ms/step - loss: 0.0249 - accuracy:
0.9895 - val_loss: 1.4418 - val_accuracy: 0.7330
Epoch 35/100
63/63 [==============================] - 5s 73ms/step - loss: 0.0092 - accuracy:
0.9980 - val_loss: 1.6587 - val_accuracy: 0.7420
Epoch 36/100
63/63 [==============================] - 4s 61ms/step - loss: 0.0200 - accuracy:
0.9935 - val_loss: 1.6532 - val_accuracy: 0.7350
Epoch 37/100
63/63 [==============================] - 4s 67ms/step - loss: 0.0344 - accuracy:
0.9890 - val_loss: 1.5939 - val_accuracy: 0.7270
Epoch 38/100
63/63 [==============================] - 8s 113ms/step - loss: 0.0581 - accurac
y: 0.9790 - val_loss: 1.6121 - val_accuracy: 0.7250
Epoch 39/100
63/63 [==============================] - 4s 59ms/step - loss: 0.0595 - accuracy:
0.9800 - val_loss: 1.3743 - val_accuracy: 0.7460
Epoch 40/100
63/63 [==============================] - 4s 63ms/step - loss: 0.0412 - accuracy:
0.9855 - val_loss: 1.4092 - val_accuracy: 0.7390
Epoch 41/100
63/63 [==============================] - 7s 100ms/step - loss: 0.0101 - accurac
y: 0.9990 - val_loss: 1.5602 - val_accuracy: 0.7410
Epoch 42/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0070 - accuracy:
0.9965 - val_loss: 1.5401 - val_accuracy: 0.7510
Epoch 43/100
63/63 [==============================] - 4s 64ms/step - loss: 0.0071 - accuracy:
0.9970 - val_loss: 1.5454 - val_accuracy: 0.7450
Epoch 44/100
63/63 [==============================] - 6s 90ms/step - loss: 0.0048 - accuracy:
0.9985 - val_loss: 1.8561 - val_accuracy: 0.7490
Epoch 45/100
63/63 [==============================] - 4s 65ms/step - loss: 0.0282 - accuracy:
0.9925 - val_loss: 1.5158 - val_accuracy: 0.7300
Epoch 46/100
63/63 [==============================] - 7s 103ms/step - loss: 0.0311 - accurac
y: 0.9860 - val_loss: 1.6100 - val_accuracy: 0.7310
Epoch 47/100
63/63 [==============================] - 4s 60ms/step - loss: 0.0403 - accuracy:
0.9885 - val_loss: 1.7674 - val_accuracy: 0.7290
Epoch 48/100
63/63 [==============================] - 6s 86ms/step - loss: 0.0850 - accuracy:
0.9695 - val_loss: 1.4952 - val_accuracy: 0.7260
Epoch 49/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0217 - accuracy:
0.9935 - val_loss: 1.6123 - val_accuracy: 0.7540
Epoch 50/100
63/63 [==============================] - 5s 70ms/step - loss: 0.0596 - accuracy:
0.9790 - val_loss: 1.6671 - val_accuracy: 0.7320
Epoch 51/100
63/63 [==============================] - 6s 89ms/step - loss: 0.0266 - accuracy:
0.9910 - val_loss: 1.4271 - val_accuracy: 0.7320
Epoch 52/100
63/63 [==============================] - 4s 59ms/step - loss: 0.0124 - accuracy:
0.9970 - val_loss: 1.5567 - val_accuracy: 0.7410
Epoch 53/100
63/63 [==============================] - 4s 60ms/step - loss: 0.0131 - accuracy:
0.9960 - val_loss: 1.6237 - val_accuracy: 0.7350
Epoch 54/100
```

```
63/63 [==============================] - 7s 109ms/step - loss: 0.0099 - accurac
y: 0.9970 - val_loss: 1.9384 - val_accuracy: 0.7490
Epoch 55/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0078 - accuracy:
0.9970 - val_loss: 1.8897 - val_accuracy: 0.7400
Epoch 56/100
63/63 [==============================] - 4s 61ms/step - loss: 0.0509 - accuracy:
0.9845 - val_loss: 1.4919 - val_accuracy: 0.7540
Epoch 57/100
63/63 [==============================] - 6s 85ms/step - loss: 0.0423 - accuracy:
0.9870 - val_loss: 1.7296 - val_accuracy: 0.7390
Epoch 58/100
63/63 [==============================] - 6s 85ms/step - loss: 0.0496 - accuracy:
0.9800 - val_loss: 1.6552 - val_accuracy: 0.7480
Epoch 59/100
63/63 [==============================] - 4s 57ms/step - loss: 0.0154 - accuracy:
0.9945 - val_loss: 1.6204 - val_accuracy: 0.7630
Epoch 60/100
63/63 [==============================] - 5s 83ms/step - loss: 0.0092 - accuracy:
0.9955 - val_loss: 1.8833 - val_accuracy: 0.7480
Epoch 61/100
63/63 [==============================] - 6s 88ms/step - loss: 0.0055 - accuracy:
0.9980 - val_loss: 1.9954 - val_accuracy: 0.7460
Epoch 62/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0120 - accuracy:
0.9945 - val_loss: 1.8193 - val_accuracy: 0.7420
Epoch 63/100
63/63 [==============================] - 7s 110ms/step - loss: 0.0260 - accurac
y: 0.9900 - val_loss: 1.7549 - val_accuracy: 0.7290
Epoch 64/100
63/63 [==============================] - 4s 61ms/step - loss: 0.0134 - accuracy:
0.9935 - val_loss: 1.9858 - val_accuracy: 0.7470
Epoch 65/100
63/63 [==============================] - 6s 94ms/step - loss: 0.0134 - accuracy:
0.9955 - val_loss: 1.9928 - val_accuracy: 0.7470
Epoch 66/100
63/63 [==============================] - 4s 59ms/step - loss: 0.0196 - accuracy:
0.9920 - val_loss: 2.3151 - val_accuracy: 0.7200
Epoch 67/100
63/63 [==============================] - 4s 63ms/step - loss: 0.0376 - accuracy:
0.9890 - val_loss: 2.0969 - val_accuracy: 0.7330
Epoch 68/100
63/63 [==============================] - 6s 92ms/step - loss: 0.0565 - accuracy:
0.9835 - val_loss: 1.8236 - val_accuracy: 0.7450
Epoch 69/100
63/63 [==============================] - 4s 62ms/step - loss: 0.0216 - accuracy:
0.9915 - val_loss: 2.1254 - val_accuracy: 0.7390
Epoch 70/100
63/63 [==============================] - 4s 57ms/step - loss: 0.0167 - accuracy:
0.9975 - val_loss: 1.9186 - val_accuracy: 0.7540
Epoch 71/100
63/63 [==============================] - 7s 97ms/step - loss: 0.0129 - accuracy:
0.9965 - val_loss: 2.3241 - val_accuracy: 0.7460
Epoch 72/100
63/63 [==============================] - 4s 59ms/step - loss: 0.0337 - accuracy:
0.9885 - val_loss: 1.9772 - val_accuracy: 0.7320
Epoch 73/100
63/63 [==============================] - 4s 62ms/step - loss: 0.0376 - accuracy:
0.9875 - val_loss: 1.6871 - val_accuracy: 0.7460
Epoch 74/100
63/63 [==============================] - 7s 102ms/step - loss: 0.0185 - accurac
y: 0.9950 - val_loss: 1.6424 - val_accuracy: 0.7650
Epoch 75/100
63/63 [==============================] - 5s 72ms/step - loss: 0.0063 - accuracy:
0.9975 - val_loss: 1.7204 - val_accuracy: 0.7490
```

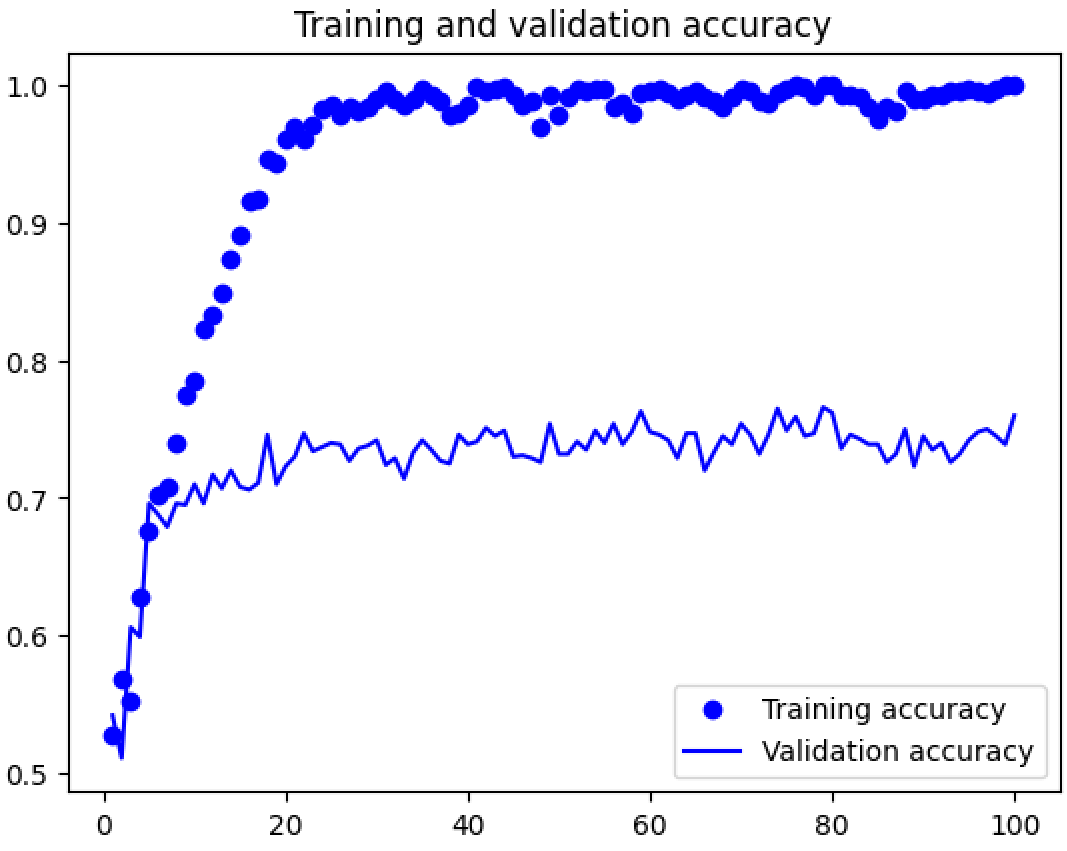
```
Epoch 76/100
63/63 [==============================] - 4s 60ms/step - loss: 0.0027 - accuracy:
0.9995 - val_loss: 1.8014 - val_accuracy: 0.7590
Epoch 77/100
63/63 [==============================] - 4s 67ms/step - loss: 0.0030 - accuracy:
0.9990 - val_loss: 2.3094 - val_accuracy: 0.7450
Epoch 78/100
63/63 [==============================] - 6s 95ms/step - loss: 0.0222 - accuracy:
0.9925 - val_loss: 1.7489 - val_accuracy: 0.7470
Epoch 79/100
63/63 [==============================] - 4s 60ms/step - loss: 0.0031 - accuracy:
0.9995 - val_loss: 1.8535 - val_accuracy: 0.7660
Epoch 80/100
63/63 [==============================] - 5s 83ms/step - loss: 0.0031 - accuracy:
0.9995 - val_loss: 1.8355 - val_accuracy: 0.7620
Epoch 81/100
63/63 [==============================] - 6s 84ms/step - loss: 0.0218 - accuracy:
0.9930 - val_loss: 2.5810 - val_accuracy: 0.7360
Epoch 82/100
63/63 [==============================] - 4s 63ms/step - loss: 0.0231 - accuracy:
0.9930 - val_loss: 2.1916 - val_accuracy: 0.7460
Epoch 83/100
63/63 [==============================] - 6s 94ms/step - loss: 0.0243 - accuracy:
0.9920 - val_loss: 2.0734 - val_accuracy: 0.7430
Epoch 84/100
63/63 [==============================] - 4s 60ms/step - loss: 0.0513 - accuracy:
0.9840 - val_loss: 1.7625 - val_accuracy: 0.7390
Epoch 85/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0618 - accuracy:
0.9755 - val_loss: 1.7235 - val_accuracy: 0.7390
Epoch 86/100
63/63 [==============================] - 6s 97ms/step - loss: 0.0514 - accuracy:
0.9835 - val_loss: 1.8148 - val_accuracy: 0.7260
Epoch 87/100
63/63 [==============================] - 4s 60ms/step - loss: 0.0623 - accuracy:
0.9820 - val_loss: 1.9512 - val_accuracy: 0.7320
Epoch 88/100
63/63 [==============================] - 5s 83ms/step - loss: 0.0103 - accuracy:
0.9965 - val_loss: 1.6809 - val_accuracy: 0.7500
Epoch 89/100
63/63 [==============================] - 4s 60ms/step - loss: 0.0252 - accuracy:
0.9900 - val_loss: 2.0125 - val_accuracy: 0.7230
Epoch 90/100
63/63 [==============================] - 4s 60ms/step - loss: 0.0235 - accuracy:
0.9905 - val_loss: 2.1302 - val_accuracy: 0.7450
Epoch 91/100
63/63 [==============================] - 7s 104ms/step - loss: 0.0201 - accurac
y: 0.9935 - val_loss: 1.9929 - val_accuracy: 0.7350
Epoch 92/100
63/63 [==============================] - 5s 68ms/step - loss: 0.0298 - accuracy:
0.9925 - val_loss: 2.0379 - val_accuracy: 0.7400
Epoch 93/100
63/63 [==============================] - 4s 60ms/step - loss: 0.0153 - accuracy:
0.9960 - val_loss: 2.1076 - val_accuracy: 0.7260
Epoch 94/100
63/63 [==============================] - 5s 83ms/step - loss: 0.0162 - accuracy:
0.9960 - val_loss: 1.8447 - val_accuracy: 0.7320
Epoch 95/100
63/63 [==============================] - 6s 87ms/step - loss: 0.0106 - accuracy:
0.9975 - val_loss: 2.0605 - val_accuracy: 0.7420
Epoch 96/100
63/63 [==============================] - 4s 59ms/step - loss: 0.0131 - accuracy:
0.9955 - val_loss: 1.7317 - val_accuracy: 0.7480
Epoch 97/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0185 - accuracy:
```

```
0.9950 - val_loss: 2.0382 - val_accuracy: 0.7500
Epoch 98/100
63/63 [==============================] - 7s 101ms/step - loss: 0.0045 - accurac
y: 0.9980 - val_loss: 2.0909 - val_accuracy: 0.7450
Epoch 99/100
63/63 [==============================] - 4s 58ms/step - loss: 0.0032 - accuracy:
0.9995 - val_loss: 2.1063 - val_accuracy: 0.7390
Epoch 100/100
63/63 [==============================] - 4s 59ms/step - loss: 0.0015 - accuracy:
0.9995 - val_loss: 1.9841 - val_accuracy: 0.7600
```

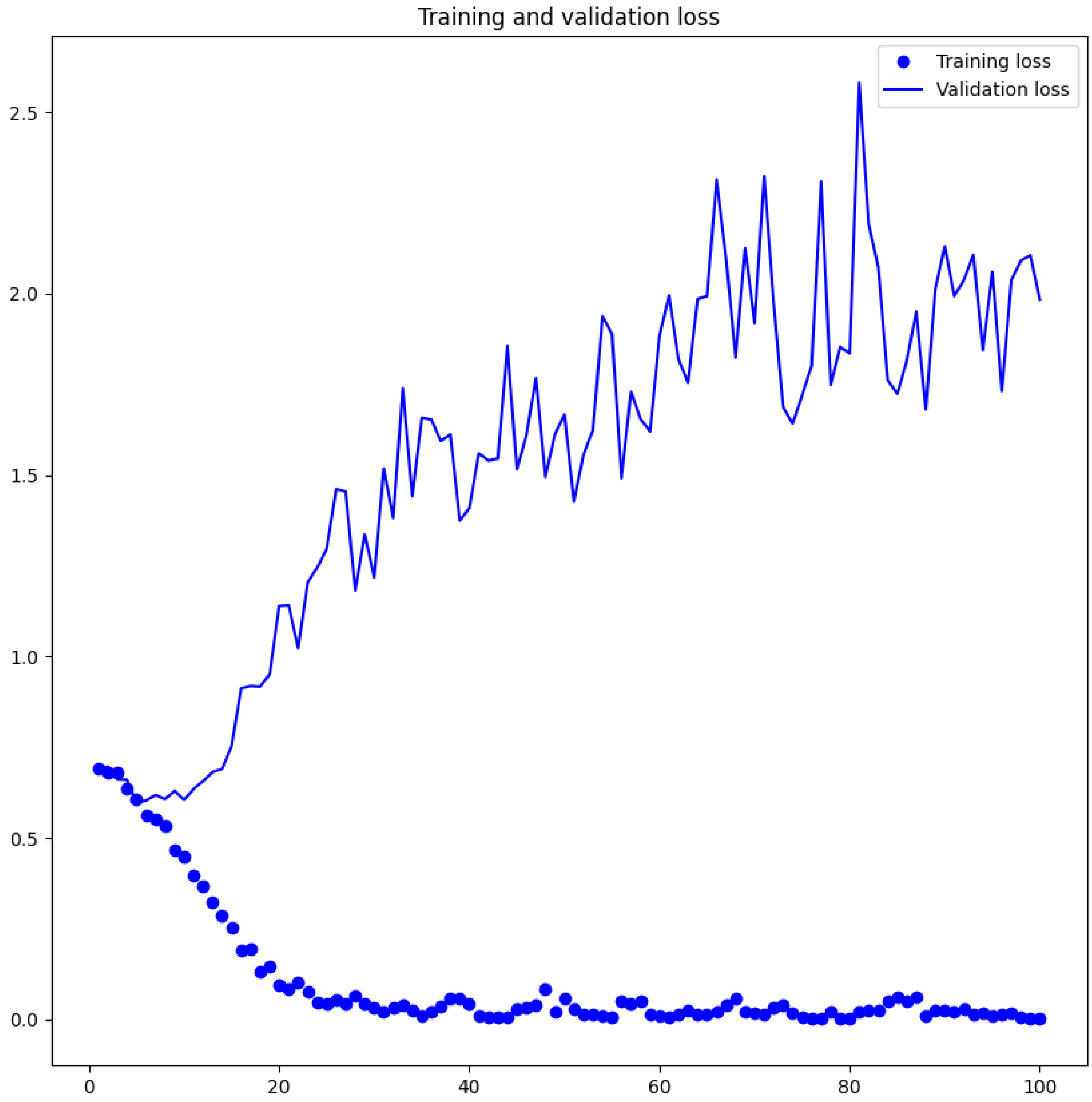
Curves of loss and accuracy during training

In [16]:

```python
import matplotlib.pyplot as plt
accuracy = history_5555.history["accuracy"]
val_accuracy = history_5555.history["val_accuracy"]
loss = history_5555.history["loss"]
val_loss = history_5555.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.figure(figsize=(10, 10))
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()
```



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



Test Accuracy of model

In [17]:

```
test5555 = keras.models.load_model("convnet_from_scratch.keras")
test_loss, test_acc = test5555.evaluate(test_data)
print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [==============================] - 2s 51ms/step - loss: 0.5848 - accuracy:
0.6880
Test accuracy: 0.688
```

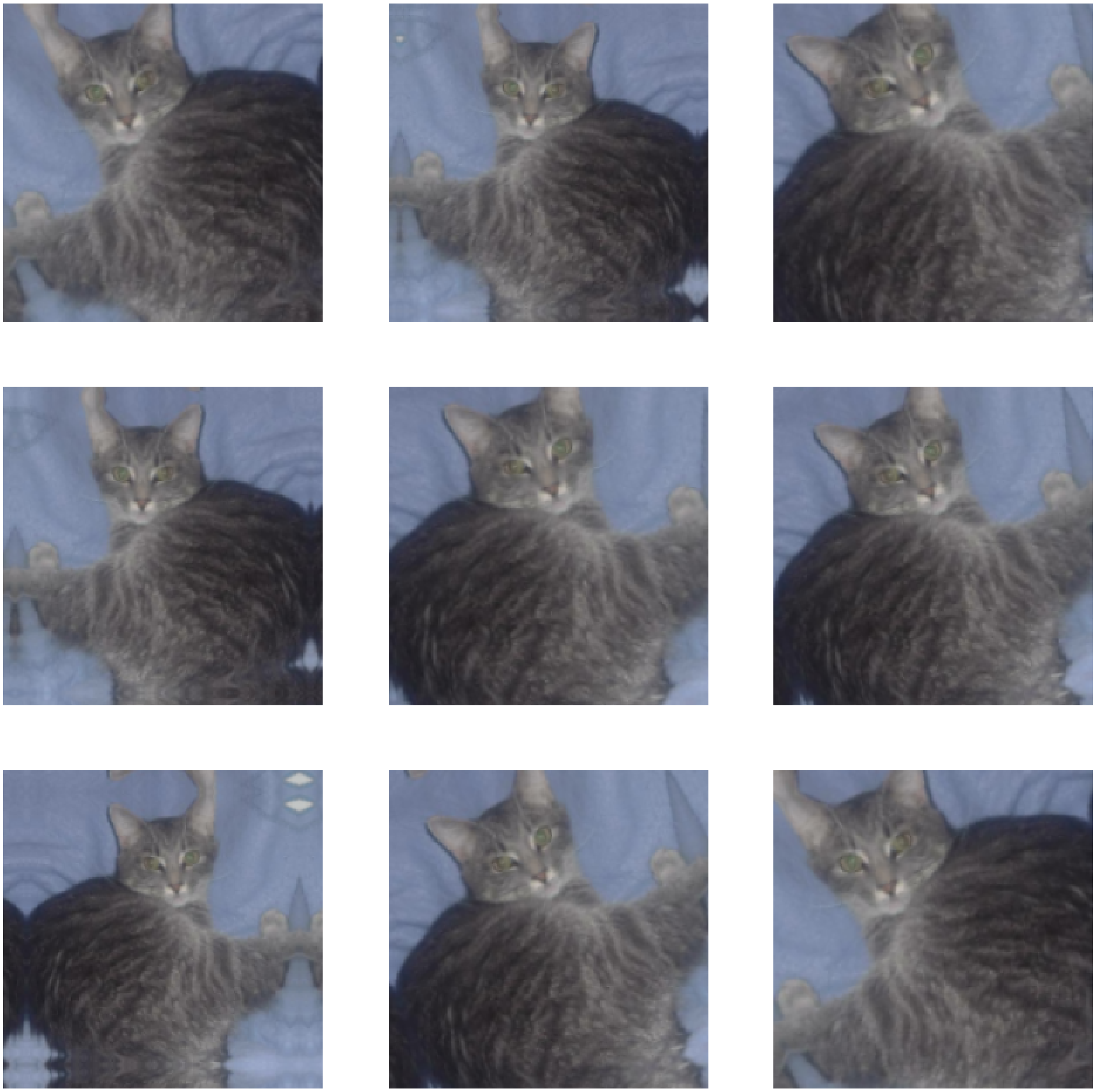
Q2. Increase your training sample size. You may pick any amount. Keep the validation and test samples the same as above. Optimize your network (again training from scratch). What performance did you achieve?

Using data augmentation

In [18]:

```
shutil.rmtree("./cats_vs_dogs_small_Q2", ignore_errors=True)
org_dir_5 = pathlib.Path("train")
new_base_dir_5 = pathlib.Path("cats_vs_dogs_small_Q2")
def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = new_base_dir_5 / 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=org_dir_5 / fname,
            dst=dir / fname)

make_subset("train", start_index=667, end_index=2167)
make_subset("validation", start_index=2168, end_index=2668)
make_subset("test", start_index=2669, end_index=3168)
augmentation_info = keras.Sequential(
[
layers.RandomFlip("horizontal"),
layers.RandomRotation(0.1),
layers.RandomZoom(0.2),
]
)
plt.figure(figsize=(10, 10))
for images, _ in train_data.take(1):
    for i in range(9):
        augmented_images = augmentation_info(images)
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(augmented_images[0].numpy().astype("uint8"))
        plt.axis("off")
```



Convolutional neural network with dropout and picture augmentation

In [19]:

```
input15 = keras.Input(shape=(180, 180, 3))
dat2 = augmentation_info(input15)
dat2 = layers.Rescaling(1./255)(dat2)
dat2 = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(dat2)
dat2 = layers.MaxPooling2D(pool_size=2)(dat2)
dat2 = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(dat2)
dat2 = layers.MaxPooling2D(pool_size=2)(dat2)
dat2 = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(dat2)
dat2 = layers.MaxPooling2D(pool_size=2)(dat2)
dat2 = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(dat2)
dat2 = layers.MaxPooling2D(pool_size=2)(dat2)
dat2 = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(dat2)
dat2 = layers.Flatten()(dat2)
dat2 = layers.Dropout(0.5)(dat2)
output15 = layers.Dense(1, activation="sigmoid")(dat2)
model15 = keras.Model(inputs=input15, outputs=output15)
model15.compile(loss="binary_crossentropy",
optimizer="adam",
metrics=["accuracy"])
callback15 = [
keras.callbacks.ModelCheckpoint(
filepath="convnet_from_scratch_with_augmentation_info.keras",
save_best_only=True,
monitor="val_loss")
]
hist15 = model15.fit(
train_data,
epochs=100,
validation_data=valid_data,
callbacks=callback15)
```

```
Epoch 1/100
63/63 [==============================] - 6s 64ms/step - loss: 0.6940 - accuracy:
0.4995 - val_loss: 0.6918 - val_accuracy: 0.5000
Epoch 2/100
63/63 [==============================] - 5s 83ms/step - loss: 0.6945 - accuracy:
0.5325 - val_loss: 0.6929 - val_accuracy: 0.5000
Epoch 3/100
63/63 [==============================] - 4s 60ms/step - loss: 0.6935 - accuracy:
0.5020 - val_loss: 0.6929 - val_accuracy: 0.5000
Epoch 4/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.4895 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 5/100
63/63 [==============================] - 7s 110ms/step - loss: 0.6942 - accurac
y: 0.5115 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 6/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6935 - accuracy:
0.4860 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 7/100
63/63 [==============================] - 4s 62ms/step - loss: 0.6925 - accuracy:
0.5100 - val_loss: 0.6913 - val_accuracy: 0.5020
Epoch 8/100
63/63 [==============================] - 6s 90ms/step - loss: 0.6934 - accuracy:
0.4965 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 9/100
63/63 [==============================] - 6s 89ms/step - loss: 0.6931 - accuracy:
0.5020 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 10/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6933 - accuracy:
0.4990 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 11/100
63/63 [==============================] - 5s 71ms/step - loss: 0.6932 - accuracy:
0.4970 - val_loss: 0.6931 - val_accuracy: 0.5070
Epoch 12/100
63/63 [==============================] - 6s 94ms/step - loss: 0.6935 - accuracy:
0.4875 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 13/100
63/63 [==============================] - 4s 62ms/step - loss: 0.6932 - accuracy:
0.5100 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 14/100
63/63 [==============================] - 7s 103ms/step - loss: 0.6933 - accurac
y: 0.4955 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 15/100
63/63 [==============================] - 5s 70ms/step - loss: 0.6934 - accuracy:
0.4775 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 16/100
63/63 [==============================] - 4s 60ms/step - loss: 0.6932 - accuracy:
0.4910 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 17/100
63/63 [==============================] - 6s 87ms/step - loss: 0.6933 - accuracy:
```

```
0.4775 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 18/100
63/63 [==============================] - 4s 61ms/step - loss: 0.6932 - accuracy:
0.4745 - val_loss: 0.6931 - val_accuracy: 0.5050
Epoch 19/100
63/63 [==============================] - 4s 62ms/step - loss: 0.6932 - accuracy:
0.5020 - val_loss: 0.6932 - val_accuracy: 0.4940
Epoch 20/100
63/63 [==============================] - 7s 96ms/step - loss: 0.6927 - accuracy:
0.5080 - val_loss: 0.6939 - val_accuracy: 0.5120
Epoch 21/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6935 - accuracy:
0.4940 - val_loss: 0.6931 - val_accuracy: 0.5320
Epoch 22/100
63/63 [==============================] - 6s 87ms/step - loss: 0.6935 - accuracy:
0.4915 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 23/100
63/63 [==============================] - 6s 82ms/step - loss: 0.6932 - accuracy:
0.4830 - val_loss: 0.6930 - val_accuracy: 0.5390
Epoch 24/100
63/63 [==============================] - 4s 60ms/step - loss: 0.6935 - accuracy:
0.5060 - val_loss: 0.6890 - val_accuracy: 0.5350
Epoch 25/100
63/63 [==============================] - 5s 74ms/step - loss: 0.6934 - accuracy:
0.4870 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 26/100
63/63 [==============================] - 6s 96ms/step - loss: 0.6933 - accuracy:
0.4955 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 27/100
63/63 [==============================] - 4s 60ms/step - loss: 0.6931 - accuracy:
0.5090 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 28/100
63/63 [==============================] - 6s 94ms/step - loss: 0.6931 - accuracy:
0.4915 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 29/100
63/63 [==============================] - 6s 82ms/step - loss: 0.6933 - accuracy:
0.4860 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 30/100
63/63 [==============================] - 4s 58ms/step - loss: 0.6931 - accuracy:
0.4985 - val_loss: 0.6931 - val_accuracy: 0.4990
Epoch 31/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6933 - accuracy:
0.4830 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 32/100
63/63 [==============================] - 8s 113ms/step - loss: 0.6932 - accurac
y: 0.5020 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 33/100
63/63 [==============================] - 4s 62ms/step - loss: 0.6932 - accuracy:
0.4925 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 34/100
63/63 [==============================] - 6s 96ms/step - loss: 0.6932 - accuracy:
0.4720 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 35/100
63/63 [==============================] - 5s 76ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 36/100
63/63 [==============================] - 6s 85ms/step - loss: 0.6932 - accuracy:
0.4820 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 37/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6932 - accuracy:
0.4920 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 38/100
63/63 [==============================] - 4s 62ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 39/100
```

```
63/63 [==============================] - 7s 101ms/step - loss: 0.6932 - accurac
y: 0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 40/100
63/63 [==============================] - 4s 61ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 41/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 42/100
63/63 [==============================] - 6s 89ms/step - loss: 0.6932 - accuracy:
0.4940 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 43/100
63/63 [==============================] - 6s 84ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 44/100
63/63 [==============================] - 4s 61ms/step - loss: 0.6932 - accuracy:
0.4870 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 45/100
63/63 [==============================] - 8s 118ms/step - loss: 0.6932 - accurac
y: 0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 46/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6932 - accuracy:
0.4950 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 47/100
63/63 [==============================] - 5s 81ms/step - loss: 0.6932 - accuracy:
0.4820 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 48/100
63/63 [==============================] - 6s 90ms/step - loss: 0.6932 - accuracy:
0.4900 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 49/100
63/63 [==============================] - 4s 61ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 50/100
63/63 [==============================] - 7s 105ms/step - loss: 0.6932 - accurac
y: 0.4890 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 51/100
63/63 [==============================] - 4s 60ms/step - loss: 0.6932 - accuracy:
0.4860 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 52/100
63/63 [==============================] - 5s 71ms/step - loss: 0.6932 - accuracy:
0.4850 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 53/100
63/63 [==============================] - 7s 103ms/step - loss: 0.6932 - accurac
y: 0.4870 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 54/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.4910 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 55/100
63/63 [==============================] - 5s 75ms/step - loss: 0.6932 - accuracy:
0.4820 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 56/100
63/63 [==============================] - 6s 92ms/step - loss: 0.6932 - accuracy:
0.4690 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 57/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6932 - accuracy:
0.4880 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 58/100
63/63 [==============================] - 4s 61ms/step - loss: 0.6932 - accuracy:
0.4830 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 59/100
63/63 [==============================] - 6s 98ms/step - loss: 0.6932 - accuracy:
0.4760 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 60/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.4910 - val_loss: 0.6931 - val_accuracy: 0.5000
```

```
Epoch 61/100
63/63 [==============================] - 6s 94ms/step - loss: 0.6932 - accuracy:
0.4870 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 62/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.4780 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 63/100
63/63 [==============================] - 5s 69ms/step - loss: 0.6932 - accuracy:
0.4800 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 64/100
63/63 [==============================] - 6s 90ms/step - loss: 0.6932 - accuracy:
0.4810 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 65/100
63/63 [==============================] - 4s 62ms/step - loss: 0.6932 - accuracy:
0.4820 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 66/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 67/100
63/63 [==============================] - 8s 114ms/step - loss: 0.6932 - accurac
y: 0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 68/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 69/100
63/63 [==============================] - 4s 61ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 70/100
63/63 [==============================] - 6s 99ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 71/100
63/63 [==============================] - 4s 58ms/step - loss: 0.6932 - accuracy:
0.4810 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 72/100
63/63 [==============================] - 6s 88ms/step - loss: 0.6932 - accuracy:
0.4790 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 73/100
63/63 [==============================] - 6s 87ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 74/100
63/63 [==============================] - 4s 64ms/step - loss: 0.6932 - accuracy:
0.4870 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 75/100
63/63 [==============================] - 6s 94ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 76/100
63/63 [==============================] - 4s 60ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 77/100
63/63 [==============================] - 6s 96ms/step - loss: 0.6932 - accuracy:
0.4890 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 78/100
63/63 [==============================] - 5s 75ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 79/100
63/63 [==============================] - 5s 73ms/step - loss: 0.6932 - accuracy:
0.4860 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 80/100
63/63 [==============================] - 6s 95ms/step - loss: 0.6932 - accuracy:
0.4850 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 81/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6932 - accuracy:
0.4870 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 82/100
63/63 [==============================] - 7s 112ms/step - loss: 0.6932 - accurac
```

```
y: 0.4820 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 83/100
63/63 [==============================] - 4s 60ms/step - loss: 0.6932 - accuracy:
0.4790 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 84/100
63/63 [==============================] - 4s 65ms/step - loss: 0.6932 - accuracy:
0.4870 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 85/100
63/63 [==============================] - 6s 95ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 86/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.4860 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 87/100
63/63 [==============================] - 4s 61ms/step - loss: 0.6932 - accuracy:
0.4900 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 88/100
63/63 [==============================] - 6s 93ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 89/100
63/63 [==============================] - 4s 62ms/step - loss: 0.6932 - accuracy:
0.4860 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 90/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.4950 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 91/100
63/63 [==============================] - 6s 89ms/step - loss: 0.6932 - accuracy:
0.4860 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 92/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.4790 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 93/100
63/63 [==============================] - 4s 67ms/step - loss: 0.6932 - accuracy:
0.4860 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 94/100
63/63 [==============================] - 7s 113ms/step - loss: 0.6932 - accurac
y: 0.4770 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 95/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.4900 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 96/100
63/63 [==============================] - 5s 71ms/step - loss: 0.6932 - accuracy:
0.4880 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 97/100
63/63 [==============================] - 7s 97ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 98/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6932 - accuracy:
0.4780 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 99/100
63/63 [==============================] - 4s 58ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 100/100
63/63 [==============================] - 6s 95ms/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
```

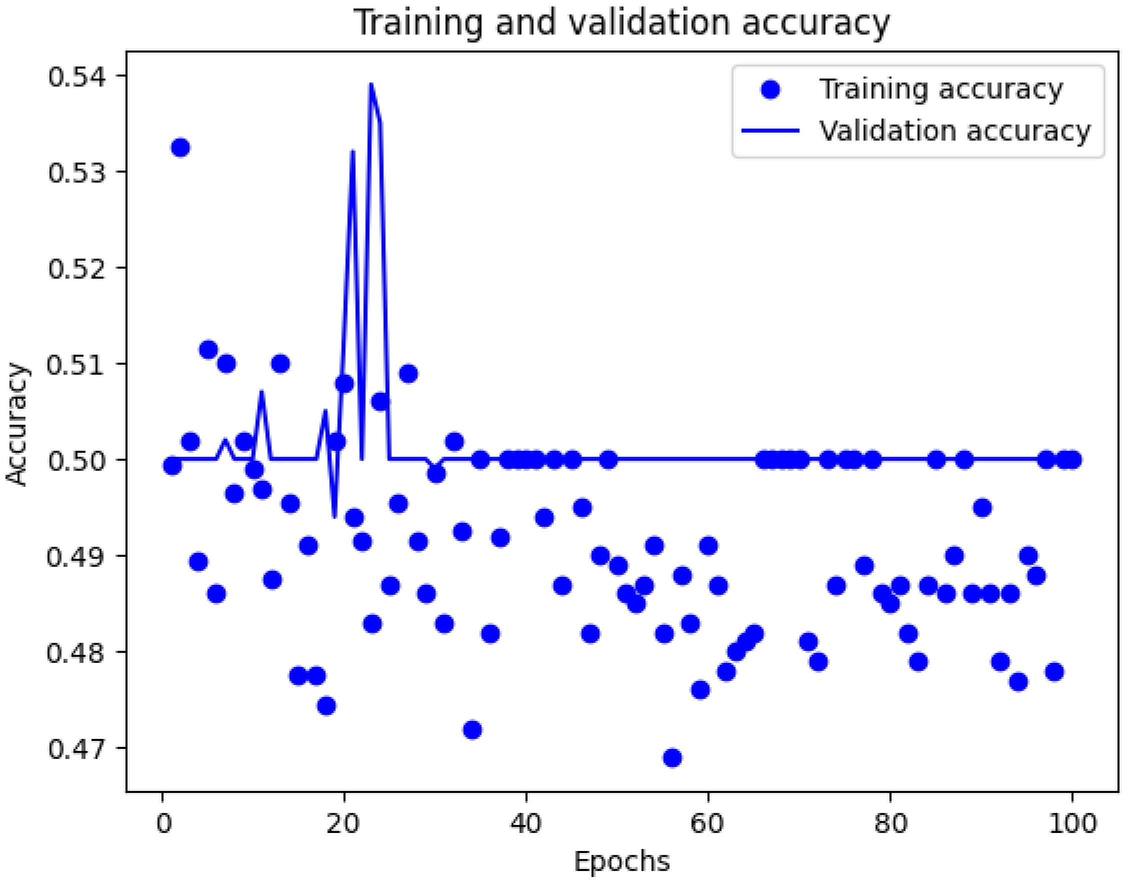
Curves of loss and accuracy during training were constructed

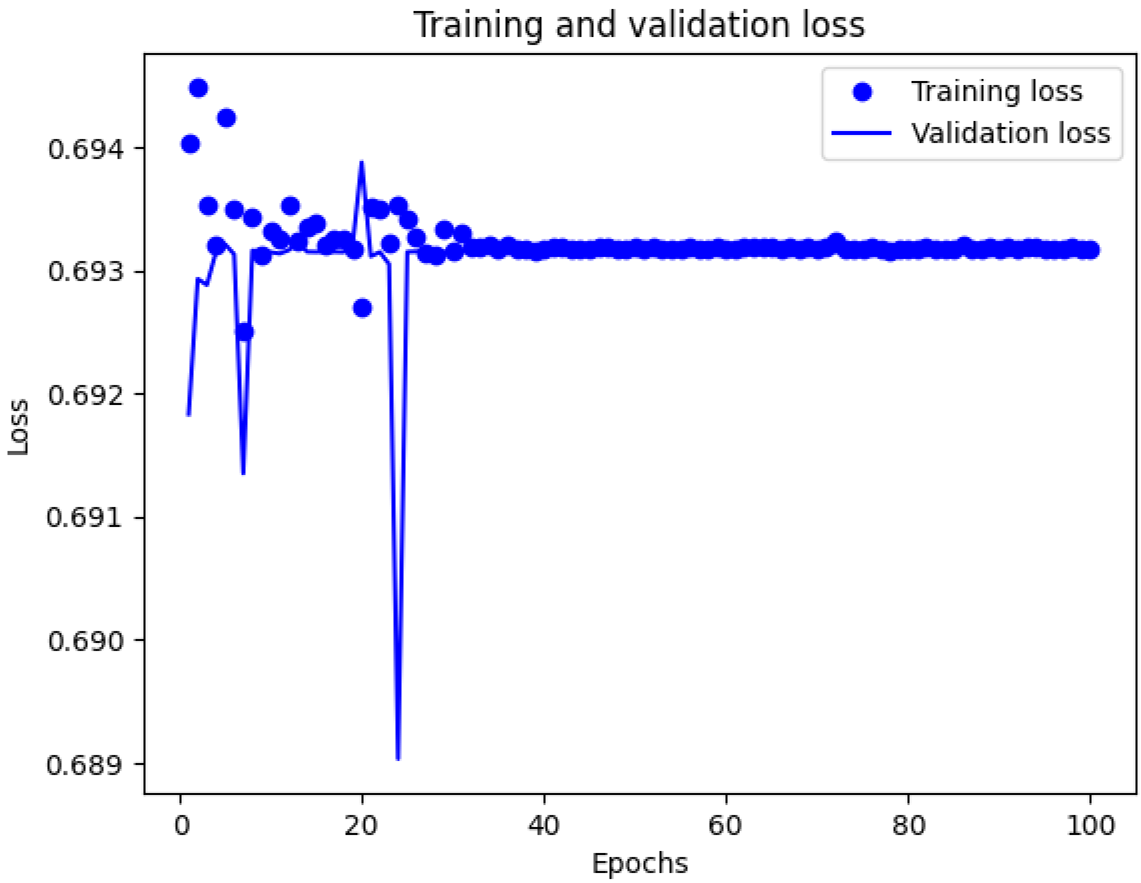
In [20]:

```
accu55 = hist15.history["accuracy"]
val55 = hist15.history["val_accuracy"]
loss55 = hist15.history["loss"]
val_loss55 = hist15.history["val_loss"]
epochs = range(1, len(accu55) + 1)
```

```
plt.plot(epochs, accu55, "bo", label="Training accuracy")
plt.plot(epochs, val55, "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, loss55, "bo", label="Training loss")
plt.plot(epochs, val_loss55, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```





Test Accuracy of model

In [21]:

```
test55accu = keras.models.load_model(
"convnet_from_scratch_with_augmentation_info.keras")
test_loss, test_acc = test55accu.evaluate(test_data)
print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [==============================] - 1s 28ms/step - loss: 0.6892 - accuracy:
0.5460
Test accuracy: 0.546
```

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 [23]:

```
new_base_dir_5 = pathlib.Path("cats_vs_dogs_small_Q3")
def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = new_base_dir_5 / 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=org_dir_5 / 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)
input20 = keras.Input(shape=(180, 180, 3))
dat3 = augmentation_info(input20)
dat3 = layers.Rescaling(1./255)(dat3)
dat3 = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(dat3)
dat3 = layers.MaxPooling2D(pool_size=2)(dat3)
dat3 = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(dat3)
dat3 = layers.MaxPooling2D(pool_size=2)(dat3)
dat3 = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(dat3)
dat3 = layers.MaxPooling2D(pool_size=2)(dat3)
dat3 = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(dat3)
dat3 = layers.MaxPooling2D(pool_size=2)(dat3)
dat3 = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(dat3)
dat3 = layers.Flatten()(dat3)
dat3 = layers.Dropout(0.5)(dat3)
output205 = layers.Dense(1, activation="sigmoid")(dat3)
mode205 = keras.Model(inputs=input20, outputs=output205)
mode205.compile(loss="binary_crossentropy",
optimizer="adam",
metrics=["accuracy"])
callback20 = [
keras.callbacks.ModelCheckpoint(
filepath="convnet_from_scratch_with_augmentation_info.keras",
save_best_only=True,
monitor="val_loss")
]
hist205 = mode205.fit(
train_data,
epochs=100,
validation_data=valid_data,
callbacks=callback20)
```

```
Epoch 1/100
63/63 [==============================] - 9s 108ms/step - loss: 0.6972 - accurac
y: 0.5010 - val_loss: 0.7019 - val_accuracy: 0.5000
Epoch 2/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6937 - accuracy:
0.4970 - val_loss: 0.6929 - val_accuracy: 0.5570
Epoch 3/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6932 - accuracy:
0.4935 - val_loss: 0.6901 - val_accuracy: 0.5400
Epoch 4/100
63/63 [==============================] - 7s 99ms/step - loss: 0.6888 - accuracy:
0.5390 - val_loss: 0.6853 - val_accuracy: 0.5460
Epoch 5/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6851 - accuracy:
0.5490 - val_loss: 0.6898 - val_accuracy: 0.5200
Epoch 6/100
63/63 [==============================] - 5s 83ms/step - loss: 0.6750 - accuracy:
0.5645 - val_loss: 0.6678 - val_accuracy: 0.5900
Epoch 7/100
63/63 [==============================] - 6s 92ms/step - loss: 0.6790 - accuracy:
0.5805 - val_loss: 0.6664 - val_accuracy: 0.6020
Epoch 8/100
63/63 [==============================] - 4s 62ms/step - loss: 0.6673 - accuracy:
0.5965 - val_loss: 0.7171 - val_accuracy: 0.6090
Epoch 9/100
63/63 [==============================] - 6s 92ms/step - loss: 0.6637 - accuracy:
0.6085 - val_loss: 0.6601 - val_accuracy: 0.6260
Epoch 10/100
63/63 [==============================] - 6s 96ms/step - loss: 0.6493 - accuracy:
```

```
0.6305 - val_loss: 0.6502 - val_accuracy: 0.6450
Epoch 11/100
63/63 [==============================] - 4s 59ms/step - loss: 0.6532 - accuracy:
0.6345 - val_loss: 0.6759 - val_accuracy: 0.5960
Epoch 12/100
63/63 [==============================] - 4s 63ms/step - loss: 0.6239 - accuracy:
0.6570 - val_loss: 0.7933 - val_accuracy: 0.5730
Epoch 13/100
63/63 [==============================] - 6s 91ms/step - loss: 0.6185 - accuracy:
0.6620 - val_loss: 0.6624 - val_accuracy: 0.5970
Epoch 14/100
63/63 [==============================] - 5s 77ms/step - loss: 0.6116 - accuracy:
0.6685 - val_loss: 0.6131 - val_accuracy: 0.6660
Epoch 15/100
63/63 [==============================] - 4s 59ms/step - loss: 0.5908 - accuracy:
0.6770 - val_loss: 0.7295 - val_accuracy: 0.5610
Epoch 16/100
63/63 [==============================] - 6s 86ms/step - loss: 0.5916 - accuracy:
0.6700 - val_loss: 0.6198 - val_accuracy: 0.6630
Epoch 17/100
63/63 [==============================] - 4s 66ms/step - loss: 0.5720 - accuracy:
0.6940 - val_loss: 0.5836 - val_accuracy: 0.7160
Epoch 18/100
63/63 [==============================] - 6s 87ms/step - loss: 0.5688 - accuracy:
0.6985 - val_loss: 0.5774 - val_accuracy: 0.7190
Epoch 19/100
63/63 [==============================] - 6s 85ms/step - loss: 0.5495 - accuracy:
0.7225 - val_loss: 0.6599 - val_accuracy: 0.6810
Epoch 20/100
63/63 [==============================] - 4s 62ms/step - loss: 0.5575 - accuracy:
0.7240 - val_loss: 0.5643 - val_accuracy: 0.7260
Epoch 21/100
63/63 [==============================] - 7s 100ms/step - loss: 0.5288 - accurac
y: 0.7395 - val_loss: 0.5689 - val_accuracy: 0.7170
Epoch 22/100
63/63 [==============================] - 4s 59ms/step - loss: 0.5364 - accuracy:
0.7365 - val_loss: 0.5841 - val_accuracy: 0.6940
Epoch 23/100
63/63 [==============================] - 4s 63ms/step - loss: 0.5280 - accuracy:
0.7405 - val_loss: 0.5627 - val_accuracy: 0.7030
Epoch 24/100
63/63 [==============================] - 7s 106ms/step - loss: 0.5030 - accurac
y: 0.7550 - val_loss: 0.5874 - val_accuracy: 0.7020
Epoch 25/100
63/63 [==============================] - 5s 67ms/step - loss: 0.4996 - accuracy:
0.7550 - val_loss: 0.6066 - val_accuracy: 0.7410
Epoch 26/100
63/63 [==============================] - 4s 60ms/step - loss: 0.4866 - accuracy:
0.7790 - val_loss: 0.5348 - val_accuracy: 0.7290
Epoch 27/100
63/63 [==============================] - 5s 79ms/step - loss: 0.4828 - accuracy:
0.7570 - val_loss: 0.5100 - val_accuracy: 0.7550
Epoch 28/100
63/63 [==============================] - 6s 91ms/step - loss: 0.4533 - accuracy:
0.7930 - val_loss: 0.5499 - val_accuracy: 0.7510
Epoch 29/100
63/63 [==============================] - 4s 59ms/step - loss: 0.4700 - accuracy:
0.7710 - val_loss: 0.5488 - val_accuracy: 0.7410
Epoch 30/100
63/63 [==============================] - 4s 61ms/step - loss: 0.4454 - accuracy:
0.7850 - val_loss: 0.4850 - val_accuracy: 0.7730
Epoch 31/100
63/63 [==============================] - 6s 98ms/step - loss: 0.4332 - accuracy:
0.7850 - val_loss: 0.5800 - val_accuracy: 0.7540
Epoch 32/100
```

```
63/63 [==============================] - 4s 61ms/step - loss: 0.4277 - accuracy:
0.7965 - val_loss: 0.5139 - val_accuracy: 0.7750
Epoch 33/100
63/63 [==============================] - 5s 74ms/step - loss: 0.4313 - accuracy:
0.7960 - val_loss: 0.5251 - val_accuracy: 0.7560
Epoch 34/100
63/63 [==============================] - 6s 93ms/step - loss: 0.4164 - accuracy:
0.8140 - val_loss: 0.5392 - val_accuracy: 0.7730
Epoch 35/100
63/63 [==============================] - 4s 63ms/step - loss: 0.4017 - accuracy:
0.8185 - val_loss: 0.4882 - val_accuracy: 0.7770
Epoch 36/100
63/63 [==============================] - 4s 59ms/step - loss: 0.3869 - accuracy:
0.8190 - val_loss: 0.5351 - val_accuracy: 0.7720
Epoch 37/100
63/63 [==============================] - 7s 111ms/step - loss: 0.3802 - accurac
y: 0.8310 - val_loss: 0.5031 - val_accuracy: 0.7730
Epoch 38/100
63/63 [==============================] - 4s 62ms/step - loss: 0.3654 - accuracy:
0.8365 - val_loss: 0.4818 - val_accuracy: 0.7870
Epoch 39/100
63/63 [==============================] - 5s 77ms/step - loss: 0.3626 - accuracy:
0.8350 - val_loss: 0.5019 - val_accuracy: 0.7950
Epoch 40/100
63/63 [==============================] - 6s 95ms/step - loss: 0.3613 - accuracy:
0.8365 - val_loss: 0.4913 - val_accuracy: 0.7760
Epoch 41/100
63/63 [==============================] - 4s 59ms/step - loss: 0.3413 - accuracy:
0.8475 - val_loss: 0.4924 - val_accuracy: 0.8050
Epoch 42/100
63/63 [==============================] - 4s 60ms/step - loss: 0.3396 - accuracy:
0.8460 - val_loss: 0.5486 - val_accuracy: 0.7880
Epoch 43/100
63/63 [==============================] - 7s 106ms/step - loss: 0.3487 - accurac
y: 0.8485 - val_loss: 0.5291 - val_accuracy: 0.7840
Epoch 44/100
63/63 [==============================] - 4s 61ms/step - loss: 0.3499 - accuracy:
0.8415 - val_loss: 0.4518 - val_accuracy: 0.8080
Epoch 45/100
63/63 [==============================] - 5s 74ms/step - loss: 0.3203 - accuracy:
0.8640 - val_loss: 0.5163 - val_accuracy: 0.7850
Epoch 46/100
63/63 [==============================] - 6s 91ms/step - loss: 0.2902 - accuracy:
0.8650 - val_loss: 0.4886 - val_accuracy: 0.8130
Epoch 47/100
63/63 [==============================] - 4s 60ms/step - loss: 0.3103 - accuracy:
0.8645 - val_loss: 0.5881 - val_accuracy: 0.7820
Epoch 48/100
63/63 [==============================] - 4s 59ms/step - loss: 0.2890 - accuracy:
0.8750 - val_loss: 0.4947 - val_accuracy: 0.8090
Epoch 49/100
63/63 [==============================] - 7s 108ms/step - loss: 0.2860 - accurac
y: 0.8820 - val_loss: 0.5385 - val_accuracy: 0.8060
Epoch 50/100
63/63 [==============================] - 4s 60ms/step - loss: 0.3096 - accuracy:
0.8640 - val_loss: 0.4805 - val_accuracy: 0.8110
Epoch 51/100
63/63 [==============================] - 4s 65ms/step - loss: 0.2644 - accuracy:
0.8955 - val_loss: 0.4768 - val_accuracy: 0.8040
Epoch 52/100
63/63 [==============================] - 6s 88ms/step - loss: 0.2771 - accuracy:
0.8845 - val_loss: 0.5528 - val_accuracy: 0.8050
Epoch 53/100
63/63 [==============================] - 6s 81ms/step - loss: 0.2677 - accuracy:
0.8735 - val_loss: 0.4809 - val_accuracy: 0.8060
```

```
Epoch 54/100
63/63 [==============================] - 6s 86ms/step - loss: 0.2395 - accuracy:
0.9050 - val_loss: 0.5428 - val_accuracy: 0.7870
Epoch 55/100
63/63 [==============================] - 6s 94ms/step - loss: 0.2577 - accuracy:
0.8890 - val_loss: 0.5717 - val_accuracy: 0.7860
Epoch 56/100
63/63 [==============================] - 4s 62ms/step - loss: 0.2621 - accuracy:
0.8905 - val_loss: 0.4615 - val_accuracy: 0.8190
Epoch 57/100
63/63 [==============================] - 4s 60ms/step - loss: 0.2333 - accuracy:
0.9040 - val_loss: 0.5694 - val_accuracy: 0.8020
Epoch 58/100
63/63 [==============================] - 7s 100ms/step - loss: 0.2450 - accurac
y: 0.9020 - val_loss: 0.5687 - val_accuracy: 0.8040
Epoch 59/100
63/63 [==============================] - 4s 64ms/step - loss: 0.2414 - accuracy:
0.9005 - val_loss: 0.5420 - val_accuracy: 0.8000
Epoch 60/100
63/63 [==============================] - 5s 84ms/step - loss: 0.2198 - accuracy:
0.9095 - val_loss: 0.5349 - val_accuracy: 0.8080
Epoch 61/100
63/63 [==============================] - 6s 91ms/step - loss: 0.2454 - accuracy:
0.9050 - val_loss: 0.5594 - val_accuracy: 0.8110
Epoch 62/100
63/63 [==============================] - 4s 60ms/step - loss: 0.2235 - accuracy:
0.9075 - val_loss: 0.5179 - val_accuracy: 0.8250
Epoch 63/100
63/63 [==============================] - 7s 111ms/step - loss: 0.2333 - accurac
y: 0.9030 - val_loss: 0.4688 - val_accuracy: 0.8260
Epoch 64/100
63/63 [==============================] - 4s 60ms/step - loss: 0.2064 - accuracy:
0.9185 - val_loss: 0.4936 - val_accuracy: 0.8240
Epoch 65/100
63/63 [==============================] - 5s 84ms/step - loss: 0.2138 - accuracy:
0.9130 - val_loss: 0.5039 - val_accuracy: 0.8340
Epoch 66/100
63/63 [==============================] - 4s 60ms/step - loss: 0.2119 - accuracy:
0.9065 - val_loss: 0.4969 - val_accuracy: 0.8300
Epoch 67/100
63/63 [==============================] - 4s 64ms/step - loss: 0.2093 - accuracy:
0.9115 - val_loss: 0.5490 - val_accuracy: 0.8070
Epoch 68/100
63/63 [==============================] - 8s 112ms/step - loss: 0.1940 - accurac
y: 0.9210 - val_loss: 0.5922 - val_accuracy: 0.8180
Epoch 69/100
63/63 [==============================] - 4s 59ms/step - loss: 0.2011 - accuracy:
0.9180 - val_loss: 0.7071 - val_accuracy: 0.8140
Epoch 70/100
63/63 [==============================] - 6s 94ms/step - loss: 0.2061 - accuracy:
0.9130 - val_loss: 0.5459 - val_accuracy: 0.8100
Epoch 71/100
63/63 [==============================] - 4s 65ms/step - loss: 0.1647 - accuracy:
0.9360 - val_loss: 0.5432 - val_accuracy: 0.8190
Epoch 72/100
63/63 [==============================] - 6s 84ms/step - loss: 0.1855 - accuracy:
0.9235 - val_loss: 0.5144 - val_accuracy: 0.8270
Epoch 73/100
63/63 [==============================] - 6s 95ms/step - loss: 0.1764 - accuracy:
0.9240 - val_loss: 0.6365 - val_accuracy: 0.8080
Epoch 74/100
63/63 [==============================] - 4s 60ms/step - loss: 0.2007 - accuracy:
0.9200 - val_loss: 0.5216 - val_accuracy: 0.8180
Epoch 75/100
63/63 [==============================] - 4s 62ms/step - loss: 0.1641 - accuracy:
```

```
0.9285 - val_loss: 0.5169 - val_accuracy: 0.8280
Epoch 76/100
63/63 [==============================] - 6s 98ms/step - loss: 0.1646 - accuracy:
0.9295 - val_loss: 0.5807 - val_accuracy: 0.8310
Epoch 77/100
63/63 [==============================] - 4s 67ms/step - loss: 0.1709 - accuracy:
0.9310 - val_loss: 0.5933 - val_accuracy: 0.8290
Epoch 78/100
63/63 [==============================] - 5s 73ms/step - loss: 0.1785 - accuracy:
0.9195 - val_loss: 0.5593 - val_accuracy: 0.8240
Epoch 79/100
63/63 [==============================] - 6s 92ms/step - loss: 0.1504 - accuracy:
0.9410 - val_loss: 0.5441 - val_accuracy: 0.8320
Epoch 80/100
63/63 [==============================] - 4s 59ms/step - loss: 0.1520 - accuracy:
0.9420 - val_loss: 0.5859 - val_accuracy: 0.8220
Epoch 81/100
63/63 [==============================] - 4s 59ms/step - loss: 0.1522 - accuracy:
0.9355 - val_loss: 0.5924 - val_accuracy: 0.8390
Epoch 82/100
63/63 [==============================] - 7s 103ms/step - loss: 0.1567 - accurac
y: 0.9325 - val_loss: 0.6619 - val_accuracy: 0.8220
Epoch 83/100
63/63 [==============================] - 4s 66ms/step - loss: 0.1435 - accuracy:
0.9415 - val_loss: 0.6384 - val_accuracy: 0.8280
Epoch 84/100
63/63 [==============================] - 6s 85ms/step - loss: 0.2077 - accuracy:
0.9245 - val_loss: 0.5266 - val_accuracy: 0.8250
Epoch 85/100
63/63 [==============================] - 6s 88ms/step - loss: 0.1530 - accuracy:
0.9365 - val_loss: 0.6318 - val_accuracy: 0.8120
Epoch 86/100
63/63 [==============================] - 4s 59ms/step - loss: 0.1402 - accuracy:
0.9440 - val_loss: 0.5595 - val_accuracy: 0.8360
Epoch 87/100
63/63 [==============================] - 4s 68ms/step - loss: 0.1408 - accuracy:
0.9410 - val_loss: 0.6616 - val_accuracy: 0.8150
Epoch 88/100
63/63 [==============================] - 6s 95ms/step - loss: 0.1508 - accuracy:
0.9405 - val_loss: 0.5933 - val_accuracy: 0.8190
Epoch 89/100
63/63 [==============================] - 4s 60ms/step - loss: 0.1482 - accuracy:
0.9450 - val_loss: 0.5836 - val_accuracy: 0.8210
Epoch 90/100
63/63 [==============================] - 6s 89ms/step - loss: 0.1113 - accuracy:
0.9580 - val_loss: 0.6371 - val_accuracy: 0.8300
Epoch 91/100
63/63 [==============================] - 6s 83ms/step - loss: 0.1172 - accuracy:
0.9510 - val_loss: 0.7138 - val_accuracy: 0.8270
Epoch 92/100
63/63 [==============================] - 4s 62ms/step - loss: 0.1322 - accuracy:
0.9505 - val_loss: 0.6161 - val_accuracy: 0.8390
Epoch 93/100
63/63 [==============================] - 5s 84ms/step - loss: 0.1253 - accuracy:
0.9505 - val_loss: 0.6556 - val_accuracy: 0.8140
Epoch 94/100
63/63 [==============================] - 4s 61ms/step - loss: 0.1312 - accuracy:
0.9445 - val_loss: 0.6108 - val_accuracy: 0.8380
Epoch 95/100
63/63 [==============================] - 4s 59ms/step - loss: 0.1129 - accuracy:
0.9595 - val_loss: 0.6639 - val_accuracy: 0.8180
Epoch 96/100
63/63 [==============================] - 7s 101ms/step - loss: 0.1171 - accurac
y: 0.9610 - val_loss: 0.6599 - val_accuracy: 0.8210
Epoch 97/100
```

```
63/63 [==============================] - 4s 61ms/step - loss: 0.1219 - accuracy:
0.9520 - val_loss: 0.6116 - val_accuracy: 0.8350
Epoch 98/100
63/63 [==============================] - 5s 80ms/step - loss: 0.1166 - accuracy:
0.9565 - val_loss: 0.6336 - val_accuracy: 0.8430
Epoch 99/100
63/63 [==============================] - 4s 60ms/step - loss: 0.1413 - accuracy:
0.9415 - val_loss: 0.6130 - val_accuracy: 0.8280
Epoch 100/100
63/63 [==============================] - 4s 63ms/step - loss: 0.1561 - accuracy:
0.9380 - val_loss: 0.5629 - val_accuracy: 0.8340
```

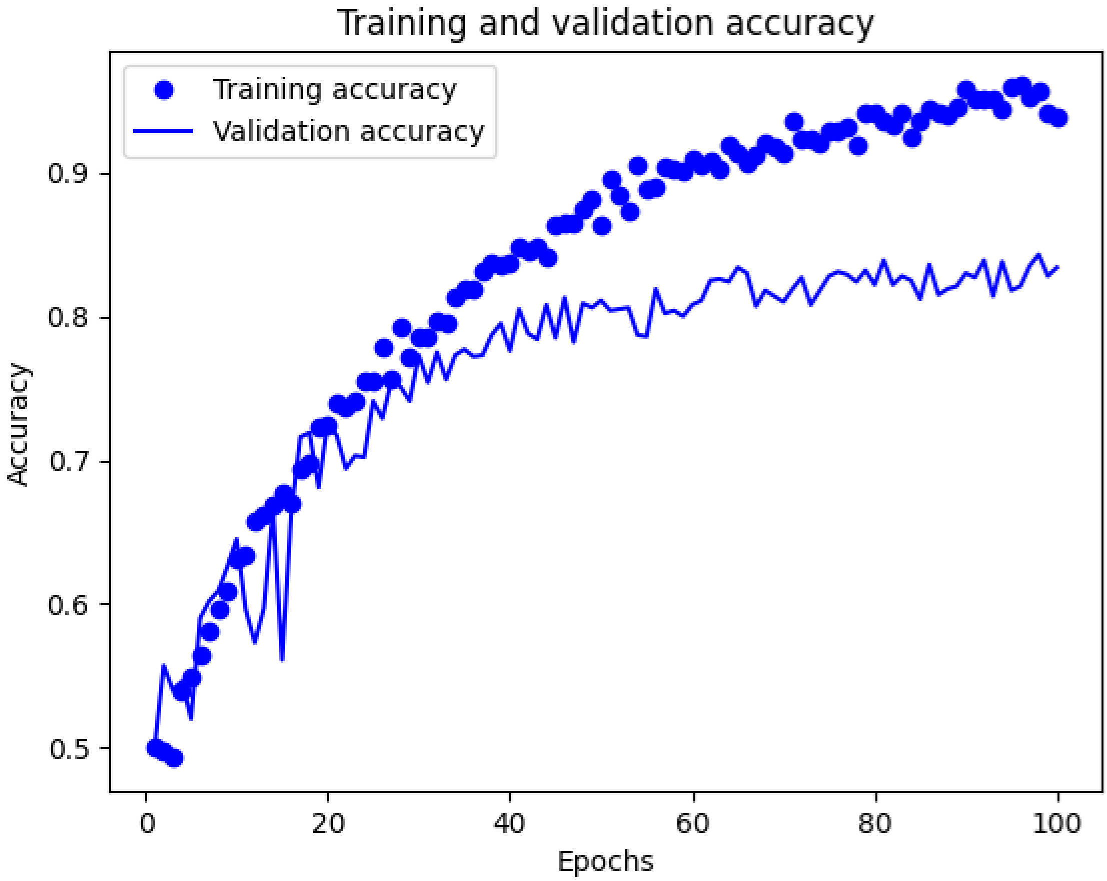
Curves of loss and accuracy during training

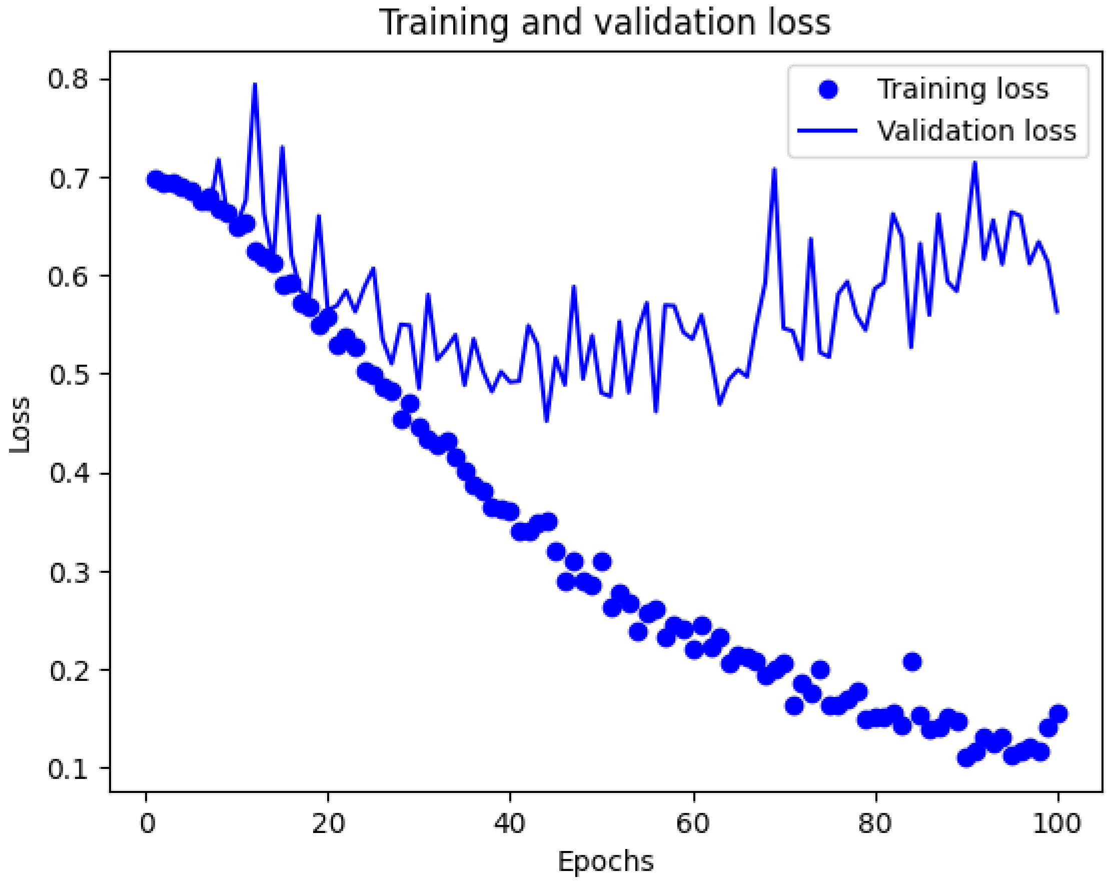
In [24]:

```
accur205 = hist205.history["accuracy"]
valac205 = hist205.history["val_accuracy"]
loss205 = hist205.history["loss"]
valloss205 = hist205.history["val_loss"]
epochs = range(1, len(accur205) + 1)

plt.plot(epochs, accur205, "bo", label="Training accuracy")
plt.plot(epochs, valac205, "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, loss205, "bo", label="Training loss")
plt.plot(epochs, valloss205, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```





Test Accuracy of model

In [25]:

```
testacc205 = keras.models.load_model(
"convnet_from_scratch_with_augmentation_info.keras")
test_loss, test_acc = testacc205.evaluate(test_data)
print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [==============================] - 1s 29ms/step - loss: 0.4388 - accuracy:
0.8010
Test accuracy: 0.801
```

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 [28]:

```
convoluted_55 = keras.applications.vgg16.VGG16(
weights="imagenet",
include_top=False,
input_shape=(180, 180, 3))
```

In [27]:

```
convoluted_55.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 [29]:

```python
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_55.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 42ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 45ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 65ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
```

```
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 3s 3s/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 2s 2s/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 22ms/step
```

```
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 27ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 23ms/step
1/1 [==============================] - 0s 20ms/step
```

In [30]:

```
train_features.shape
```

Out[30]:

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

Model Fitting

In [31]:

```
inp6_5 = keras.Input(shape=(5, 5, 512))
dat4 = layers.Flatten()(inp6_5)
dat4 = layers.Dense(256)(dat4)
dat4 = layers.Dropout(0.5)(dat4)
out4_5 = layers.Dense(1, activation="sigmoid")(dat4)
model4_5 = keras.Model(inp6_5, out4_5)
model4_5.compile(loss="binary_crossentropy",
optimizer="rmsprop",
metrics=["accuracy"])
callbac4 = [
keras.callbacks.ModelCheckpoint(
filepath="feature_extraction.keras",
save_best_only=True,
monitor="val_loss")
]
history4_5 = model4_5.fit(
train_features, train_labels,
epochs=100,
validation_data=(val_features, val_labels),
callbacks=callbac4)
```

```
Epoch 1/100
63/63 [==============================] - 2s 14ms/step - loss: 15.4367 - accurac
y: 0.9205 - val_loss: 4.1237 - val_accuracy: 0.9730
Epoch 2/100
63/63 [==============================] - 0s 7ms/step - loss: 2.0730 - accuracy:
```

```
0.9825 - val_loss: 5.5701 - val_accuracy: 0.9680
Epoch 3/100
63/63 [==============================] - 0s 8ms/step - loss: 2.1899 - accuracy:
0.9820 - val_loss: 5.7305 - val_accuracy: 0.9690
Epoch 4/100
63/63 [==============================] - 1s 8ms/step - loss: 1.7572 - accuracy:
0.9875 - val_loss: 5.8861 - val_accuracy: 0.9730
Epoch 5/100
63/63 [==============================] - 1s 8ms/step - loss: 0.5082 - accuracy:
0.9930 - val_loss: 5.3854 - val_accuracy: 0.9740
Epoch 6/100
63/63 [==============================] - 1s 8ms/step - loss: 0.6159 - accuracy:
0.9935 - val_loss: 5.9216 - val_accuracy: 0.9770
Epoch 7/100
63/63 [==============================] - 1s 9ms/step - loss: 0.4797 - accuracy:
0.9945 - val_loss: 5.3000 - val_accuracy: 0.9780
Epoch 8/100
63/63 [==============================] - 1s 8ms/step - loss: 0.7796 - accuracy:
0.9945 - val_loss: 6.3743 - val_accuracy: 0.9700
Epoch 9/100
63/63 [==============================] - 1s 8ms/step - loss: 2.1030e-35 - accura
cy: 1.0000 - val_loss: 6.3743 - val_accuracy: 0.9700
Epoch 10/100
63/63 [==============================] - 1s 8ms/step - loss: 0.7785 - accuracy:
0.9955 - val_loss: 5.3761 - val_accuracy: 0.9770
Epoch 11/100
63/63 [==============================] - 1s 8ms/step - loss: 0.2058 - accuracy:
0.9970 - val_loss: 5.5050 - val_accuracy: 0.9750
Epoch 12/100
63/63 [==============================] - 1s 8ms/step - loss: 0.2483 - accuracy:
0.9980 - val_loss: 5.8444 - val_accuracy: 0.9740
Epoch 13/100
63/63 [==============================] - 1s 8ms/step - loss: 1.5594e-09 - accura
cy: 1.0000 - val_loss: 5.8446 - val_accuracy: 0.9740
Epoch 14/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0047 - accuracy:
0.9995 - val_loss: 6.5319 - val_accuracy: 0.9710
Epoch 15/100
63/63 [==============================] - 0s 7ms/step - loss: 0.1904 - accuracy:
0.9970 - val_loss: 7.2063 - val_accuracy: 0.9730
Epoch 16/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0217 - accuracy:
0.9995 - val_loss: 6.4394 - val_accuracy: 0.9690
Epoch 17/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0482 - accuracy:
0.9985 - val_loss: 7.8953 - val_accuracy: 0.9710
Epoch 18/100
63/63 [==============================] - 0s 5ms/step - loss: 7.7250e-07 - accura
cy: 1.0000 - val_loss: 7.1013 - val_accuracy: 0.9680
Epoch 19/100
63/63 [==============================] - 0s 6ms/step - loss: 1.0476e-23 - accura
cy: 1.0000 - val_loss: 7.1013 - val_accuracy: 0.9680
Epoch 20/100
63/63 [==============================] - 0s 5ms/step - loss: 2.1492e-08 - accura
cy: 1.0000 - val_loss: 6.9281 - val_accuracy: 0.9690
Epoch 21/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9281 - val_accuracy: 0.9690
Epoch 22/100
63/63 [==============================] - 0s 5ms/step - loss: 1.1427e-32 - accura
cy: 1.0000 - val_loss: 6.9281 - val_accuracy: 0.9690
Epoch 23/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9281 - val_accuracy: 0.9690
Epoch 24/100
```

```
63/63 [==============================] - 0s 5ms/step - loss: 2.9137e-11 - accura
cy: 1.0000 - val_loss: 6.9279 - val_accuracy: 0.9690
Epoch 25/100
63/63 [==============================] - 0s 5ms/step - loss: 5.8634e-17 - accura
cy: 1.0000 - val_loss: 6.9279 - val_accuracy: 0.9690
Epoch 26/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0171 - accuracy:
0.9995 - val_loss: 6.5194 - val_accuracy: 0.9720
Epoch 27/100
63/63 [==============================] - 0s 5ms/step - loss: 0.1154 - accuracy:
0.9985 - val_loss: 6.3270 - val_accuracy: 0.9690
Epoch 28/100
63/63 [==============================] - 0s 5ms/step - loss: 0.1656 - accuracy:
0.9980 - val_loss: 5.6332 - val_accuracy: 0.9720
Epoch 29/100
63/63 [==============================] - 0s 5ms/step - loss: 2.0977e-17 - accura
cy: 1.0000 - val_loss: 5.6332 - val_accuracy: 0.9720
Epoch 30/100
63/63 [==============================] - 0s 6ms/step - loss: 0.2317 - accuracy:
0.9990 - val_loss: 6.1838 - val_accuracy: 0.9680
Epoch 31/100
63/63 [==============================] - 0s 6ms/step - loss: 2.3169e-25 - accura
cy: 1.0000 - val_loss: 6.1838 - val_accuracy: 0.9680
Epoch 32/100
63/63 [==============================] - 0s 6ms/step - loss: 2.2903e-35 - accura
cy: 1.0000 - val_loss: 6.1838 - val_accuracy: 0.9680
Epoch 33/100
63/63 [==============================] - 0s 6ms/step - loss: 1.1031e-20 - accura
cy: 1.0000 - val_loss: 6.1838 - val_accuracy: 0.9680
Epoch 34/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0532 - accuracy:
0.9990 - val_loss: 6.7912 - val_accuracy: 0.9700
Epoch 35/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.7912 - val_accuracy: 0.9700
Epoch 36/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.7912 - val_accuracy: 0.9700
Epoch 37/100
63/63 [==============================] - 0s 6ms/step - loss: 3.0875e-27 - accura
cy: 1.0000 - val_loss: 6.7912 - val_accuracy: 0.9700
Epoch 38/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0530 - accuracy:
0.9995 - val_loss: 12.0373 - val_accuracy: 0.9580
Epoch 39/100
63/63 [==============================] - 0s 5ms/step - loss: 0.2283 - accuracy:
0.9980 - val_loss: 5.5929 - val_accuracy: 0.9750
Epoch 40/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 5.5929 - val_accuracy: 0.9750
Epoch 41/100
63/63 [==============================] - 0s 6ms/step - loss: 1.0839e-06 - accura
cy: 1.0000 - val_loss: 5.7537 - val_accuracy: 0.9730
Epoch 42/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0792 - accuracy:
0.9985 - val_loss: 7.8778 - val_accuracy: 0.9650
Epoch 43/100
63/63 [==============================] - 0s 8ms/step - loss: 0.0923 - accuracy:
0.9990 - val_loss: 8.1292 - val_accuracy: 0.9680
Epoch 44/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0816 - accuracy:
0.9990 - val_loss: 7.2156 - val_accuracy: 0.9720
Epoch 45/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0200 - accuracy:
0.9995 - val_loss: 6.8722 - val_accuracy: 0.9750
```

```
Epoch 46/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0149 - accuracy:
0.9995 - val_loss: 7.6890 - val_accuracy: 0.9710
Epoch 47/100
63/63 [==============================] - 1s 9ms/step - loss: 1.2350e-06 - accura
cy: 1.0000 - val_loss: 6.9846 - val_accuracy: 0.9710
Epoch 48/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9846 - val_accuracy: 0.9710
Epoch 49/100
63/63 [==============================] - 0s 8ms/step - loss: 7.5158e-20 - accura
cy: 1.0000 - val_loss: 6.9846 - val_accuracy: 0.9710
Epoch 50/100
63/63 [==============================] - 1s 8ms/step - loss: 0.1193 - accuracy:
0.9985 - val_loss: 6.1226 - val_accuracy: 0.9710
Epoch 51/100
63/63 [==============================] - 1s 9ms/step - loss: 3.3556e-08 - accura
cy: 1.0000 - val_loss: 6.1161 - val_accuracy: 0.9710
Epoch 52/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0543 - accuracy:
0.9995 - val_loss: 8.2579 - val_accuracy: 0.9720
Epoch 53/100
63/63 [==============================] - 0s 8ms/step - loss: 0.0081 - accuracy:
0.9995 - val_loss: 6.6215 - val_accuracy: 0.9730
Epoch 54/100
63/63 [==============================] - 1s 9ms/step - loss: 8.7477e-16 - accura
cy: 1.0000 - val_loss: 6.6215 - val_accuracy: 0.9730
Epoch 55/100
63/63 [==============================] - 1s 9ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.6215 - val_accuracy: 0.9730
Epoch 56/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.6215 - val_accuracy: 0.9730
Epoch 57/100
63/63 [==============================] - 0s 8ms/step - loss: 0.0452 - accuracy:
0.9995 - val_loss: 8.9818 - val_accuracy: 0.9690
Epoch 58/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0166 - accuracy:
0.9995 - val_loss: 7.6858 - val_accuracy: 0.9700
Epoch 59/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.6858 - val_accuracy: 0.9700
Epoch 60/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.6858 - val_accuracy: 0.9700
Epoch 61/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.6858 - val_accuracy: 0.9700
Epoch 62/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.6858 - val_accuracy: 0.9700
Epoch 63/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.6858 - val_accuracy: 0.9700
Epoch 64/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.6858 - val_accuracy: 0.9700
Epoch 65/100
63/63 [==============================] - 0s 5ms/step - loss: 1.0099e-10 - accura
cy: 1.0000 - val_loss: 7.6860 - val_accuracy: 0.9700
Epoch 66/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.6860 - val_accuracy: 0.9700
Epoch 67/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accura
```

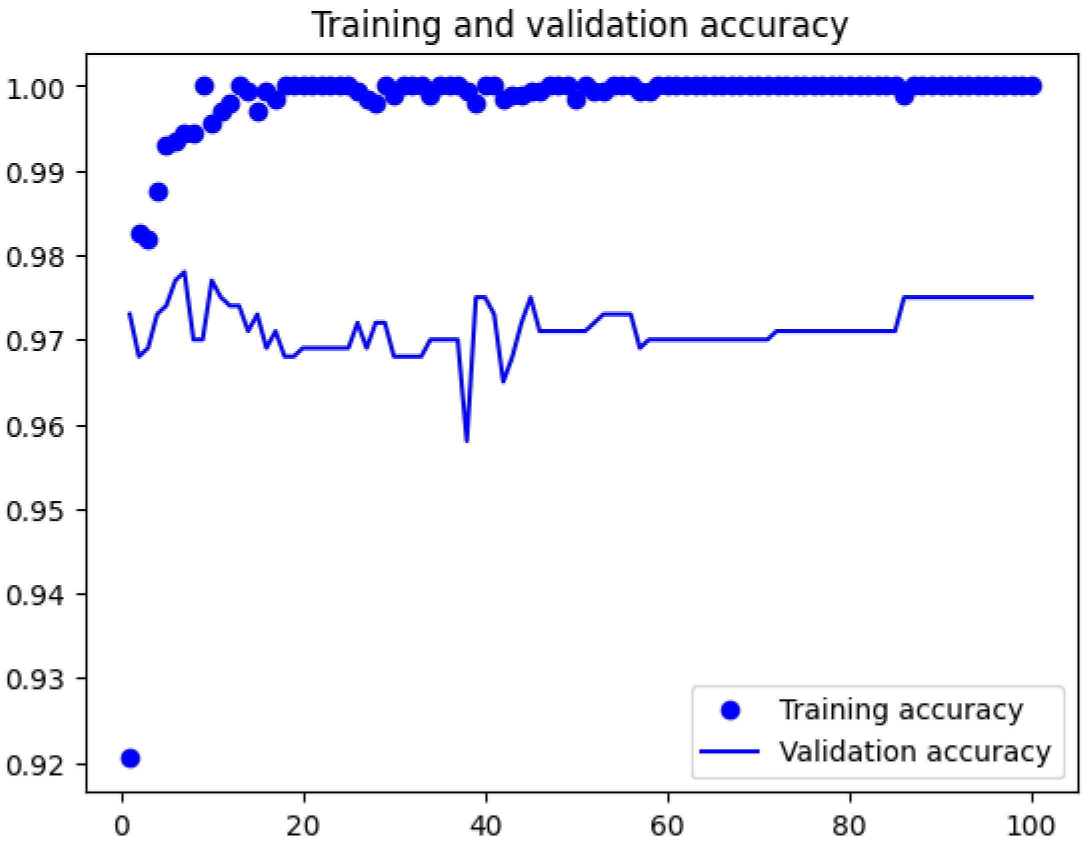
```
cy: 1.0000 - val_loss: 7.6860 - val_accuracy: 0.9700
Epoch 68/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.6860 - val_accuracy: 0.9700
Epoch 69/100
63/63 [==============================] - 0s 5ms/step - loss: 6.8027e-16 - accura
cy: 1.0000 - val_loss: 7.6860 - val_accuracy: 0.9700
Epoch 70/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.6860 - val_accuracy: 0.9700
Epoch 71/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.6860 - val_accuracy: 0.9700
Epoch 72/100
63/63 [==============================] - 0s 5ms/step - loss: 5.9340e-08 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 73/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 74/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 75/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 76/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 77/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 78/100
63/63 [==============================] - 0s 5ms/step - loss: 1.9708e-36 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 79/100
63/63 [==============================] - 0s 5ms/step - loss: 2.2019e-20 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 80/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 81/100
63/63 [==============================] - 0s 5ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 82/100
63/63 [==============================] - 0s 6ms/step - loss: 2.8166e-19 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 83/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 84/100
63/63 [==============================] - 0s 6ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 85/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 7.5013 - val_accuracy: 0.9710
Epoch 86/100
63/63 [==============================] - 0s 7ms/step - loss: 0.1254 - accuracy:
0.9990 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 87/100
63/63 [==============================] - 0s 8ms/step - loss: 5.5457e-33 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 88/100
63/63 [==============================] - 1s 8ms/step - loss: 1.2129e-17 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 89/100
```

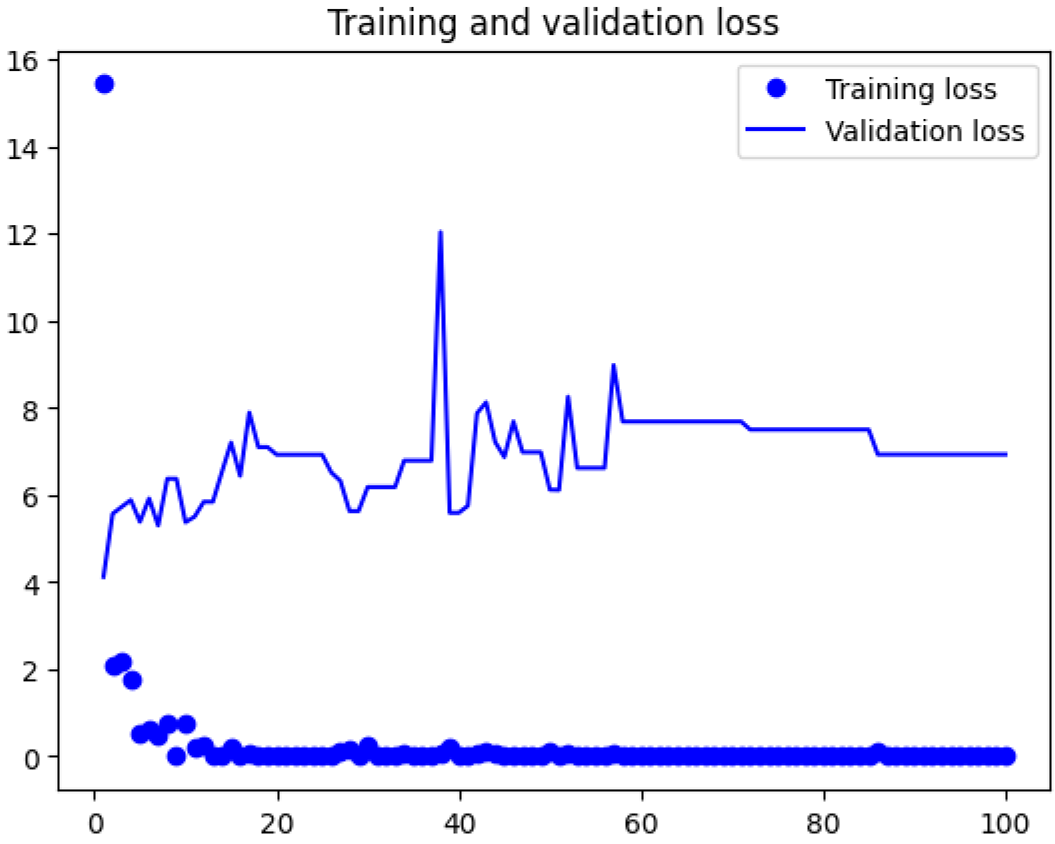
```
63/63 [==============================] - 1s 9ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 90/100
63/63 [==============================] - 1s 9ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 91/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 92/100
63/63 [==============================] - 0s 8ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 93/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 94/100
63/63 [==============================] - 0s 8ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 95/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 96/100
63/63 [==============================] - 0s 8ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 97/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 98/100
63/63 [==============================] - 0s 8ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 99/100
63/63 [==============================] - 0s 7ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
Epoch 100/100
63/63 [==============================] - 1s 8ms/step - loss: 0.0000e+00 - accura
cy: 1.0000 - val_loss: 6.9325 - val_accuracy: 0.9750
```

Curves of loss and accuracy during training

In [32]:

```
accur4_5 = history4_5.history["accuracy"]
valac4_5 = history4_5.history["val_accuracy"]
loss4_5 = history4_5.history["loss"]
val_loss4_5 = history4_5.history["val_loss"]
epochs = range(1, len(accur4_5) + 1)
plt.plot(epochs, accur4_5, "bo", label="Training accuracy")
plt.plot(epochs, valac4_5, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.legend()
plt.figure()
plt.plot(epochs, loss4_5, "bo", label="Training loss")
plt.plot(epochs, val_loss4_5, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.legend()
plt.show()
```





Freezing and Unfreezing the Pre-trained Convolutional Base

In [33]:

```
convoluted_55 = keras.applications.vgg16.VGG16(
weights="imagenet",
include_top=False)
```

```
convoluted_55.trainable = False
convoluted_55.trainable = True
print("This is the number of trainable weights "
"before freezing the conv base:", len(convoluted_55.trainable_weights))
convoluted_55.trainable = False
print("This is the number of trainable weights "
"after freezing the conv base:", len(convoluted_55.trainable_weights))
```

```
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 [34]:

```
augmented2 = keras.Sequential(
[
layers.RandomFlip("horizontal"),
layers.RandomRotation(0.1),
layers.RandomZoom(0.2),
]
)
input225 = keras.Input(shape=(180, 180, 3))
datx = augmented2(input225)
datx =keras.layers.Lambda(
lambda x: keras.applications.vgg16.preprocess_input(x))(datx)
datx = convoluted_55(datx)
datx = layers.Flatten()(datx)
datx = layers.Dense(256)(datx)
datx = layers.Dropout(0.5)(datx)
outputs = layers.Dense(1, activation="sigmoid")(datx)
modelx = keras.Model(input225, outputs)
modelx.compile(loss="binary_crossentropy",
optimizer="rmsprop",
metrics=["accuracy"])
callback55 = [
keras.callbacks.ModelCheckpoint(
filepath="features_extraction_with_augmentation2.keras",
save_best_only=True,
monitor="val_loss"
)
]
history55 = modelx.fit(
train_data,
epochs=100,
validation_data=valid_data,
callbacks=callback55
)
```

```
Epoch 1/100
63/63 [==============================] - 13s 189ms/step - loss: 22.9490 - accura
cy: 0.8960 - val_loss: 3.8072 - val_accuracy: 0.9650
Epoch 2/100
63/63 [==============================] - 10s 162ms/step - loss: 5.7219 - accurac
y: 0.9505 - val_loss: 2.8684 - val_accuracy: 0.9750
Epoch 3/100
63/63 [==============================] - 10s 148ms/step - loss: 6.3390 - accurac
y: 0.9540 - val_loss: 4.0743 - val_accuracy: 0.9710
Epoch 4/100
63/63 [==============================] - 10s 155ms/step - loss: 5.5666 - accurac
y: 0.9565 - val_loss: 5.9228 - val_accuracy: 0.9670
Epoch 5/100
63/63 [==============================] - 12s 182ms/step - loss: 5.3535 - accurac
```

```
y: 0.9585 - val_loss: 4.3886 - val_accuracy: 0.9760
Epoch 6/100
63/63 [==============================] - 10s 156ms/step - loss: 4.5945 - accurac
y: 0.9630 - val_loss: 3.5556 - val_accuracy: 0.9780
Epoch 7/100
63/63 [==============================] - 12s 184ms/step - loss: 2.3961 - accurac
y: 0.9705 - val_loss: 4.1458 - val_accuracy: 0.9750
Epoch 8/100
63/63 [==============================] - 10s 156ms/step - loss: 3.3599 - accurac
y: 0.9655 - val_loss: 3.8066 - val_accuracy: 0.9750
Epoch 9/100
63/63 [==============================] - 11s 162ms/step - loss: 3.4393 - accurac
y: 0.9715 - val_loss: 5.6323 - val_accuracy: 0.9680
Epoch 10/100
63/63 [==============================] - 10s 156ms/step - loss: 2.9333 - accurac
y: 0.9755 - val_loss: 4.1014 - val_accuracy: 0.9730
Epoch 11/100
63/63 [==============================] - 11s 164ms/step - loss: 1.9204 - accurac
y: 0.9780 - val_loss: 3.5522 - val_accuracy: 0.9740
Epoch 12/100
63/63 [==============================] - 12s 185ms/step - loss: 2.4561 - accurac
y: 0.9760 - val_loss: 3.5792 - val_accuracy: 0.9730
Epoch 13/100
63/63 [==============================] - 12s 184ms/step - loss: 2.1352 - accurac
y: 0.9795 - val_loss: 4.0318 - val_accuracy: 0.9730
Epoch 14/100
63/63 [==============================] - 10s 156ms/step - loss: 2.2281 - accurac
y: 0.9795 - val_loss: 3.7967 - val_accuracy: 0.9770
Epoch 15/100
63/63 [==============================] - 11s 161ms/step - loss: 1.9132 - accurac
y: 0.9760 - val_loss: 4.5134 - val_accuracy: 0.9730
Epoch 16/100
63/63 [==============================] - 10s 156ms/step - loss: 1.6948 - accurac
y: 0.9810 - val_loss: 3.1285 - val_accuracy: 0.9760
Epoch 17/100
63/63 [==============================] - 11s 163ms/step - loss: 1.8944 - accurac
y: 0.9835 - val_loss: 3.6205 - val_accuracy: 0.9720
Epoch 18/100
63/63 [==============================] - 12s 185ms/step - loss: 1.7545 - accurac
y: 0.9785 - val_loss: 3.4345 - val_accuracy: 0.9730
Epoch 19/100
63/63 [==============================] - 10s 163ms/step - loss: 1.9748 - accurac
y: 0.9795 - val_loss: 3.8025 - val_accuracy: 0.9700
Epoch 20/100
63/63 [==============================] - 12s 185ms/step - loss: 1.2130 - accurac
y: 0.9800 - val_loss: 3.6883 - val_accuracy: 0.9710
Epoch 21/100
63/63 [==============================] - 11s 169ms/step - loss: 1.3820 - accurac
y: 0.9805 - val_loss: 2.5172 - val_accuracy: 0.9780
Epoch 22/100
63/63 [==============================] - 10s 161ms/step - loss: 1.3886 - accurac
y: 0.9805 - val_loss: 4.8747 - val_accuracy: 0.9690
Epoch 23/100
63/63 [==============================] - 11s 165ms/step - loss: 1.1297 - accurac
y: 0.9830 - val_loss: 2.2747 - val_accuracy: 0.9740
Epoch 24/100
63/63 [==============================] - 10s 161ms/step - loss: 0.8875 - accurac
y: 0.9850 - val_loss: 2.6168 - val_accuracy: 0.9720
Epoch 25/100
63/63 [==============================] - 10s 157ms/step - loss: 0.9098 - accurac
y: 0.9875 - val_loss: 2.4353 - val_accuracy: 0.9740
Epoch 26/100
63/63 [==============================] - 11s 170ms/step - loss: 0.7248 - accurac
y: 0.9865 - val_loss: 2.2206 - val_accuracy: 0.9770
Epoch 27/100
```

```
63/63 [==============================] - 11s 165ms/step - loss: 0.9529 - accurac
y: 0.9860 - val_loss: 2.2737 - val_accuracy: 0.9740
Epoch 28/100
63/63 [==============================] - 11s 171ms/step - loss: 0.8625 - accurac
y: 0.9845 - val_loss: 2.0987 - val_accuracy: 0.9760
Epoch 29/100
63/63 [==============================] - 11s 168ms/step - loss: 0.6512 - accurac
y: 0.9890 - val_loss: 2.2131 - val_accuracy: 0.9740
Epoch 30/100
63/63 [==============================] - 11s 169ms/step - loss: 0.8512 - accurac
y: 0.9825 - val_loss: 1.9424 - val_accuracy: 0.9770
Epoch 31/100
63/63 [==============================] - 10s 157ms/step - loss: 0.9575 - accurac
y: 0.9865 - val_loss: 2.6479 - val_accuracy: 0.9740
Epoch 32/100
63/63 [==============================] - 11s 162ms/step - loss: 0.8191 - accurac
y: 0.9835 - val_loss: 2.2714 - val_accuracy: 0.9740
Epoch 33/100
63/63 [==============================] - 11s 165ms/step - loss: 0.9931 - accurac
y: 0.9815 - val_loss: 2.0729 - val_accuracy: 0.9770
Epoch 34/100
63/63 [==============================] - 11s 165ms/step - loss: 0.7865 - accurac
y: 0.9855 - val_loss: 2.1301 - val_accuracy: 0.9760
Epoch 35/100
63/63 [==============================] - 10s 161ms/step - loss: 0.5564 - accurac
y: 0.9885 - val_loss: 2.0553 - val_accuracy: 0.9770
Epoch 36/100
63/63 [==============================] - 10s 163ms/step - loss: 0.4407 - accurac
y: 0.9900 - val_loss: 1.9934 - val_accuracy: 0.9780
Epoch 37/100
63/63 [==============================] - 12s 187ms/step - loss: 0.5154 - accurac
y: 0.9905 - val_loss: 2.3211 - val_accuracy: 0.9750
Epoch 38/100
63/63 [==============================] - 11s 171ms/step - loss: 0.4658 - accurac
y: 0.9900 - val_loss: 1.5497 - val_accuracy: 0.9810
Epoch 39/100
63/63 [==============================] - 12s 186ms/step - loss: 0.4579 - accurac
y: 0.9870 - val_loss: 1.8514 - val_accuracy: 0.9800
Epoch 40/100
63/63 [==============================] - 11s 162ms/step - loss: 0.4994 - accurac
y: 0.9875 - val_loss: 1.7324 - val_accuracy: 0.9760
Epoch 41/100
63/63 [==============================] - 11s 167ms/step - loss: 0.7345 - accurac
y: 0.9855 - val_loss: 1.9861 - val_accuracy: 0.9770
Epoch 42/100
63/63 [==============================] - 11s 164ms/step - loss: 0.7831 - accurac
y: 0.9850 - val_loss: 1.6055 - val_accuracy: 0.9790
Epoch 43/100
63/63 [==============================] - 11s 166ms/step - loss: 0.4722 - accurac
y: 0.9875 - val_loss: 1.8365 - val_accuracy: 0.9760
Epoch 44/100
63/63 [==============================] - 12s 187ms/step - loss: 0.3945 - accurac
y: 0.9900 - val_loss: 2.1472 - val_accuracy: 0.9750
Epoch 45/100
63/63 [==============================] - 10s 161ms/step - loss: 0.4544 - accurac
y: 0.9890 - val_loss: 2.4467 - val_accuracy: 0.9770
Epoch 46/100
63/63 [==============================] - 11s 164ms/step - loss: 0.4293 - accurac
y: 0.9905 - val_loss: 2.0681 - val_accuracy: 0.9750
Epoch 47/100
63/63 [==============================] - 12s 186ms/step - loss: 0.4455 - accurac
y: 0.9900 - val_loss: 1.8200 - val_accuracy: 0.9780
Epoch 48/100
63/63 [==============================] - 11s 161ms/step - loss: 0.4746 - accurac
y: 0.9890 - val_loss: 1.9754 - val_accuracy: 0.9790
```

```
Epoch 49/100
63/63 [==============================] - 11s 168ms/step - loss: 0.4752 - accurac
y: 0.9905 - val_loss: 1.8041 - val_accuracy: 0.9790
Epoch 50/100
63/63 [==============================] - 11s 161ms/step - loss: 0.5457 - accurac
y: 0.9905 - val_loss: 1.9286 - val_accuracy: 0.9800
Epoch 51/100
63/63 [==============================] - 11s 167ms/step - loss: 0.3622 - accurac
y: 0.9890 - val_loss: 1.7732 - val_accuracy: 0.9790
Epoch 52/100
63/63 [==============================] - 12s 188ms/step - loss: 0.4161 - accurac
y: 0.9895 - val_loss: 1.7089 - val_accuracy: 0.9830
Epoch 53/100
63/63 [==============================] - 11s 165ms/step - loss: 0.7322 - accurac
y: 0.9875 - val_loss: 1.7472 - val_accuracy: 0.9780
Epoch 54/100
63/63 [==============================] - 11s 170ms/step - loss: 0.5380 - accurac
y: 0.9930 - val_loss: 1.5636 - val_accuracy: 0.9770
Epoch 55/100
63/63 [==============================] - 11s 164ms/step - loss: 0.6207 - accurac
y: 0.9855 - val_loss: 1.6946 - val_accuracy: 0.9760
Epoch 56/100
63/63 [==============================] - 13s 197ms/step - loss: 0.3681 - accurac
y: 0.9915 - val_loss: 1.4571 - val_accuracy: 0.9790
Epoch 57/100
63/63 [==============================] - 12s 189ms/step - loss: 0.4005 - accurac
y: 0.9875 - val_loss: 1.5301 - val_accuracy: 0.9780
Epoch 58/100
63/63 [==============================] - 12s 188ms/step - loss: 0.5922 - accurac
y: 0.9855 - val_loss: 2.2221 - val_accuracy: 0.9710
Epoch 59/100
63/63 [==============================] - 12s 196ms/step - loss: 0.3437 - accurac
y: 0.9900 - val_loss: 1.4329 - val_accuracy: 0.9760
Epoch 60/100
63/63 [==============================] - 11s 165ms/step - loss: 0.3713 - accurac
y: 0.9905 - val_loss: 1.7917 - val_accuracy: 0.9770
Epoch 61/100
63/63 [==============================] - 11s 164ms/step - loss: 0.4266 - accurac
y: 0.9895 - val_loss: 1.7367 - val_accuracy: 0.9780
Epoch 62/100
63/63 [==============================] - 12s 186ms/step - loss: 0.1004 - accurac
y: 0.9950 - val_loss: 2.0524 - val_accuracy: 0.9730
Epoch 63/100
63/63 [==============================] - 11s 162ms/step - loss: 0.5146 - accurac
y: 0.9890 - val_loss: 1.7326 - val_accuracy: 0.9780
Epoch 64/100
63/63 [==============================] - 12s 193ms/step - loss: 0.5649 - accurac
y: 0.9890 - val_loss: 1.2597 - val_accuracy: 0.9790
Epoch 65/100
63/63 [==============================] - 11s 164ms/step - loss: 0.1991 - accurac
y: 0.9955 - val_loss: 1.3488 - val_accuracy: 0.9750
Epoch 66/100
63/63 [==============================] - 10s 160ms/step - loss: 0.3836 - accurac
y: 0.9905 - val_loss: 1.4672 - val_accuracy: 0.9740
Epoch 67/100
63/63 [==============================] - 11s 163ms/step - loss: 0.3299 - accurac
y: 0.9915 - val_loss: 1.7849 - val_accuracy: 0.9760
Epoch 68/100
63/63 [==============================] - 11s 166ms/step - loss: 0.3617 - accurac
y: 0.9935 - val_loss: 3.5628 - val_accuracy: 0.9640
Epoch 69/100
63/63 [==============================] - 11s 166ms/step - loss: 0.4332 - accurac
y: 0.9910 - val_loss: 1.9323 - val_accuracy: 0.9750
Epoch 70/100
63/63 [==============================] - 11s 160ms/step - loss: 0.2839 - accurac
```

```
y: 0.9935 - val_loss: 1.6231 - val_accuracy: 0.9750
Epoch 71/100
63/63 [==============================] - 11s 165ms/step - loss: 0.4474 - accurac
y: 0.9910 - val_loss: 1.6687 - val_accuracy: 0.9760
Epoch 72/100
63/63 [==============================] - 12s 187ms/step - loss: 0.4375 - accurac
y: 0.9900 - val_loss: 1.4970 - val_accuracy: 0.9760
Epoch 73/100
63/63 [==============================] - 10s 162ms/step - loss: 0.4021 - accurac
y: 0.9900 - val_loss: 1.5000 - val_accuracy: 0.9770
Epoch 74/100
63/63 [==============================] - 10s 163ms/step - loss: 0.4396 - accurac
y: 0.9895 - val_loss: 2.7737 - val_accuracy: 0.9730
Epoch 75/100
63/63 [==============================] - 10s 159ms/step - loss: 0.2653 - accurac
y: 0.9905 - val_loss: 1.8835 - val_accuracy: 0.9760
Epoch 76/100
63/63 [==============================] - 11s 162ms/step - loss: 0.2300 - accurac
y: 0.9890 - val_loss: 1.7468 - val_accuracy: 0.9780
Epoch 77/100
63/63 [==============================] - 12s 188ms/step - loss: 0.4560 - accurac
y: 0.9875 - val_loss: 1.7949 - val_accuracy: 0.9740
Epoch 78/100
63/63 [==============================] - 11s 166ms/step - loss: 0.2564 - accurac
y: 0.9920 - val_loss: 1.9704 - val_accuracy: 0.9760
Epoch 79/100
63/63 [==============================] - 12s 187ms/step - loss: 0.2616 - accurac
y: 0.9925 - val_loss: 1.9421 - val_accuracy: 0.9750
Epoch 80/100
63/63 [==============================] - 11s 162ms/step - loss: 0.2277 - accurac
y: 0.9895 - val_loss: 2.5186 - val_accuracy: 0.9760
Epoch 81/100
63/63 [==============================] - 12s 186ms/step - loss: 0.2814 - accurac
y: 0.9945 - val_loss: 2.2089 - val_accuracy: 0.9770
Epoch 82/100
63/63 [==============================] - 10s 161ms/step - loss: 0.4812 - accurac
y: 0.9920 - val_loss: 2.1196 - val_accuracy: 0.9780
Epoch 83/100
63/63 [==============================] - 11s 163ms/step - loss: 0.3156 - accurac
y: 0.9920 - val_loss: 2.1943 - val_accuracy: 0.9780
Epoch 84/100
63/63 [==============================] - 11s 167ms/step - loss: 0.4618 - accurac
y: 0.9925 - val_loss: 1.9114 - val_accuracy: 0.9780
Epoch 85/100
63/63 [==============================] - 11s 165ms/step - loss: 0.2883 - accurac
y: 0.9950 - val_loss: 2.5847 - val_accuracy: 0.9770
Epoch 86/100
63/63 [==============================] - 11s 170ms/step - loss: 0.1928 - accurac
y: 0.9935 - val_loss: 2.3038 - val_accuracy: 0.9760
Epoch 87/100
63/63 [==============================] - 12s 191ms/step - loss: 0.4002 - accurac
y: 0.9905 - val_loss: 1.9001 - val_accuracy: 0.9780
Epoch 88/100
63/63 [==============================] - 11s 167ms/step - loss: 0.3051 - accurac
y: 0.9930 - val_loss: 2.0164 - val_accuracy: 0.9780
Epoch 89/100
63/63 [==============================] - 11s 171ms/step - loss: 0.4481 - accurac
y: 0.9925 - val_loss: 2.6404 - val_accuracy: 0.9750
Epoch 90/100
63/63 [==============================] - 12s 191ms/step - loss: 0.4309 - accurac
y: 0.9895 - val_loss: 2.0206 - val_accuracy: 0.9780
Epoch 91/100
63/63 [==============================] - 11s 168ms/step - loss: 0.2767 - accurac
y: 0.9930 - val_loss: 2.1523 - val_accuracy: 0.9750
Epoch 92/100
```

```
63/63 [==============================] - 12s 190ms/step - loss: 0.2806 - accurac
y: 0.9930 - val_loss: 3.3437 - val_accuracy: 0.9740
Epoch 93/100
63/63 [==============================] - 11s 168ms/step - loss: 0.2256 - accurac
y: 0.9945 - val_loss: 2.3427 - val_accuracy: 0.9780
Epoch 94/100
63/63 [==============================] - 11s 173ms/step - loss: 0.2519 - accurac
y: 0.9950 - val_loss: 1.9278 - val_accuracy: 0.9780
Epoch 95/100
63/63 [==============================] - 12s 192ms/step - loss: 0.2302 - accurac
y: 0.9950 - val_loss: 2.3088 - val_accuracy: 0.9760
Epoch 96/100
63/63 [==============================] - 11s 167ms/step - loss: 0.2710 - accurac
y: 0.9945 - val_loss: 2.1775 - val_accuracy: 0.9770
Epoch 97/100
63/63 [==============================] - 12s 189ms/step - loss: 0.3786 - accurac
y: 0.9915 - val_loss: 2.8941 - val_accuracy: 0.9770
Epoch 98/100
63/63 [==============================] - 11s 167ms/step - loss: 0.3222 - accurac
y: 0.9905 - val_loss: 2.6715 - val_accuracy: 0.9780
Epoch 99/100
63/63 [==============================] - 12s 190ms/step - loss: 0.2254 - accurac
y: 0.9950 - val_loss: 2.2899 - val_accuracy: 0.9790
Epoch 100/100
63/63 [==============================] - 12s 188ms/step - loss: 0.5014 - accurac
y: 0.9905 - val_loss: 2.8622 - val_accuracy: 0.9740
```

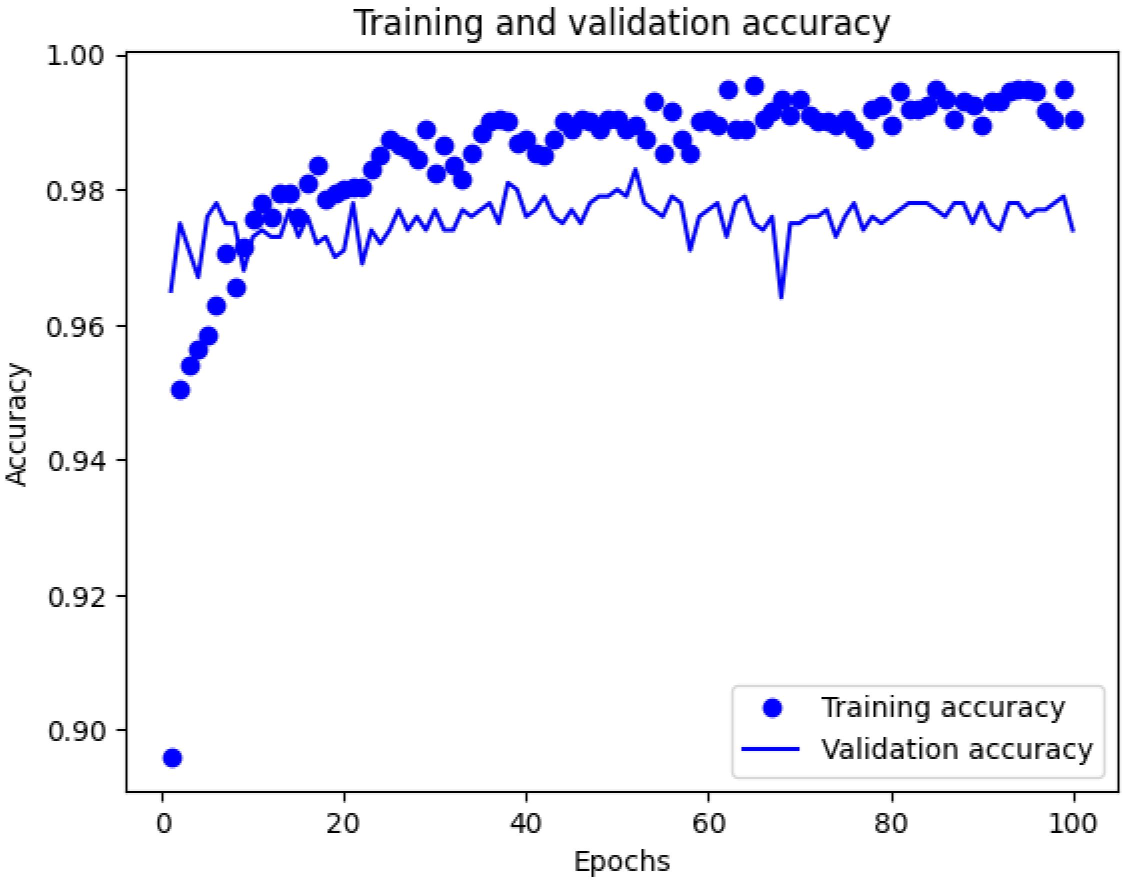
Curves of loss and accuracy during training

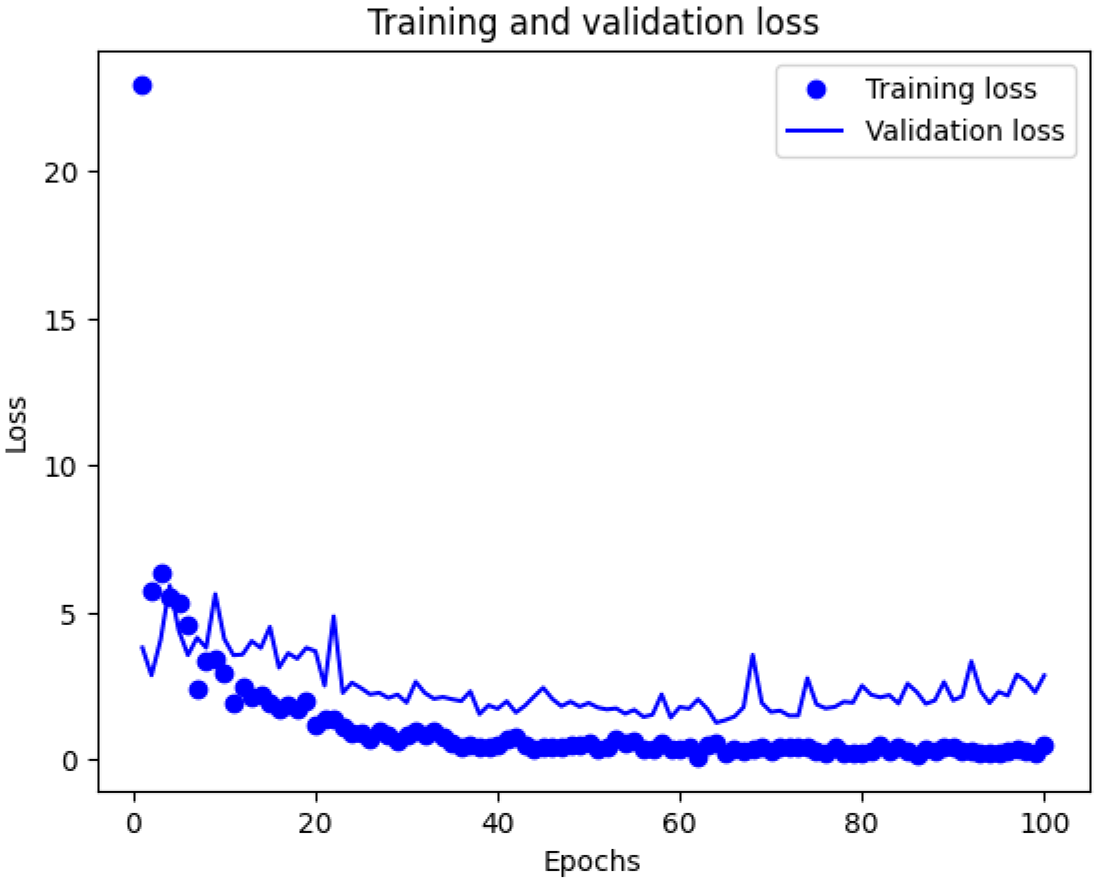
In [35]:

```
accurfi = history55.history["accuracy"]
valacfi = history55.history["val_accuracy"]
lossfi = history55.history["loss"]
vallossfi = history55.history["val_loss"]
epochs = range(1, len(accurfi) + 1)

plt.plot(epochs, accurfi, "bo", label="Training accuracy")
plt.plot(epochs, valacfi, "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, lossfi, "bo", label="Training loss")
plt.plot(epochs, vallossfi, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```





Test Accuracy of model

In [36]:

```
tesaccfi = keras.models.load_model(
"features_extraction_with_augmentation2.keras",safe_mode=False)
test_loss, test_acc = tesaccfi.evaluate(test_data)
print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [==============================] - 4s 98ms/step - loss: 2.5450 - accuracy:
0.9700
Test accuracy: 0.970
```

Fine-tuning a pretrained model

In [37]:

```
convoluted_55.trainable = True
for layer in convoluted_55.layers[:-4]:
    layer.trainable = False

modelx.compile(loss="binary_crossentropy",
optimizer=keras.optimizers.RMSprop(learning_rate=1e-5),
metrics=["accuracy"])
callbacktuning = [
keras.callbacks.ModelCheckpoint(
filepath="fine_tuning.keras",
save_best_only=True,
monitor="val_loss")
]
historytuning = modelx.fit(
train_data,
epochs=100,
validation_data=valid_data,
callbacks=callbacktuning)
```

```
Epoch 1/100
63/63 [==============================] - 14s 193ms/step - loss: 0.5540 - accurac
y: 0.9900 - val_loss: 2.2200 - val_accuracy: 0.9750
Epoch 2/100
63/63 [==============================] - 11s 178ms/step - loss: 0.3299 - accurac
y: 0.9910 - val_loss: 2.5867 - val_accuracy: 0.9740
Epoch 3/100
63/63 [==============================] - 11s 177ms/step - loss: 0.2048 - accurac
y: 0.9935 - val_loss: 2.4398 - val_accuracy: 0.9760
Epoch 4/100
63/63 [==============================] - 13s 200ms/step - loss: 0.2367 - accurac
y: 0.9935 - val_loss: 2.3321 - val_accuracy: 0.9760
Epoch 5/100
63/63 [==============================] - 13s 200ms/step - loss: 0.3934 - accurac
y: 0.9900 - val_loss: 2.2226 - val_accuracy: 0.9740
Epoch 6/100
63/63 [==============================] - 13s 201ms/step - loss: 0.7922 - accurac
y: 0.9860 - val_loss: 2.2446 - val_accuracy: 0.9720
Epoch 7/100
63/63 [==============================] - 12s 186ms/step - loss: 0.2212 - accurac
y: 0.9940 - val_loss: 1.8087 - val_accuracy: 0.9760
Epoch 8/100
63/63 [==============================] - 12s 179ms/step - loss: 0.2679 - accurac
y: 0.9950 - val_loss: 2.4060 - val_accuracy: 0.9800
Epoch 9/100
63/63 [==============================] - 11s 175ms/step - loss: 0.2669 - accurac
y: 0.9925 - val_loss: 2.1650 - val_accuracy: 0.9810
Epoch 10/100
63/63 [==============================] - 12s 188ms/step - loss: 0.1609 - accurac
y: 0.9980 - val_loss: 1.6842 - val_accuracy: 0.9810
Epoch 11/100
```

```
63/63 [==============================] - 12s 181ms/step - loss: 0.1703 - accurac
y: 0.9950 - val_loss: 2.3892 - val_accuracy: 0.9730
Epoch 12/100
63/63 [==============================] - 11s 175ms/step - loss: 0.1950 - accurac
y: 0.9950 - val_loss: 2.0507 - val_accuracy: 0.9790
Epoch 13/100
63/63 [==============================] - 12s 181ms/step - loss: 0.1503 - accurac
y: 0.9965 - val_loss: 2.2895 - val_accuracy: 0.9770
Epoch 14/100
63/63 [==============================] - 13s 202ms/step - loss: 0.2826 - accurac
y: 0.9940 - val_loss: 1.8369 - val_accuracy: 0.9800
Epoch 15/100
63/63 [==============================] - 12s 183ms/step - loss: 0.1878 - accurac
y: 0.9965 - val_loss: 2.2596 - val_accuracy: 0.9750
Epoch 16/100
63/63 [==============================] - 11s 174ms/step - loss: 0.1189 - accurac
y: 0.9960 - val_loss: 2.0028 - val_accuracy: 0.9780
Epoch 17/100
63/63 [==============================] - 12s 180ms/step - loss: 0.1812 - accurac
y: 0.9955 - val_loss: 1.8589 - val_accuracy: 0.9800
Epoch 18/100
63/63 [==============================] - 12s 185ms/step - loss: 0.1755 - accurac
y: 0.9950 - val_loss: 1.7575 - val_accuracy: 0.9770
Epoch 19/100
63/63 [==============================] - 13s 203ms/step - loss: 0.1188 - accurac
y: 0.9975 - val_loss: 2.0424 - val_accuracy: 0.9780
Epoch 20/100
63/63 [==============================] - 13s 205ms/step - loss: 0.0785 - accurac
y: 0.9970 - val_loss: 1.8716 - val_accuracy: 0.9770
Epoch 21/100
63/63 [==============================] - 12s 181ms/step - loss: 0.1470 - accurac
y: 0.9960 - val_loss: 2.0113 - val_accuracy: 0.9760
Epoch 22/100
63/63 [==============================] - 12s 189ms/step - loss: 0.1585 - accurac
y: 0.9955 - val_loss: 2.3394 - val_accuracy: 0.9760
Epoch 23/100
63/63 [==============================] - 12s 184ms/step - loss: 0.0879 - accurac
y: 0.9985 - val_loss: 2.0174 - val_accuracy: 0.9800
Epoch 24/100
63/63 [==============================] - 14s 208ms/step - loss: 0.1529 - accurac
y: 0.9965 - val_loss: 1.8208 - val_accuracy: 0.9780
Epoch 25/100
63/63 [==============================] - 12s 180ms/step - loss: 0.2570 - accurac
y: 0.9950 - val_loss: 1.7733 - val_accuracy: 0.9770
Epoch 26/100
63/63 [==============================] - 12s 181ms/step - loss: 0.1409 - accurac
y: 0.9950 - val_loss: 1.8973 - val_accuracy: 0.9790
Epoch 27/100
63/63 [==============================] - 12s 189ms/step - loss: 0.1245 - accurac
y: 0.9965 - val_loss: 1.8794 - val_accuracy: 0.9780
Epoch 28/100
63/63 [==============================] - 13s 208ms/step - loss: 0.2590 - accurac
y: 0.9965 - val_loss: 1.8682 - val_accuracy: 0.9800
Epoch 29/100
63/63 [==============================] - 12s 186ms/step - loss: 0.1471 - accurac
y: 0.9965 - val_loss: 2.1955 - val_accuracy: 0.9740
Epoch 30/100
63/63 [==============================] - 12s 182ms/step - loss: 0.2449 - accurac
y: 0.9955 - val_loss: 2.2599 - val_accuracy: 0.9780
Epoch 31/100
63/63 [==============================] - 11s 176ms/step - loss: 0.1211 - accurac
y: 0.9970 - val_loss: 2.0355 - val_accuracy: 0.9770
Epoch 32/100
63/63 [==============================] - 12s 180ms/step - loss: 0.0213 - accurac
y: 0.9985 - val_loss: 2.0327 - val_accuracy: 0.9760
```

```
Epoch 33/100
63/63 [==============================] - 11s 177ms/step - loss: 0.3052 - accurac
y: 0.9945 - val_loss: 2.0500 - val_accuracy: 0.9800
Epoch 34/100
63/63 [==============================] - 12s 182ms/step - loss: 0.1764 - accurac
y: 0.9945 - val_loss: 2.4596 - val_accuracy: 0.9780
Epoch 35/100
63/63 [==============================] - 12s 185ms/step - loss: 0.1630 - accurac
y: 0.9960 - val_loss: 1.8676 - val_accuracy: 0.9780
Epoch 36/100
63/63 [==============================] - 13s 204ms/step - loss: 0.1344 - accurac
y: 0.9960 - val_loss: 2.6449 - val_accuracy: 0.9750
Epoch 37/100
63/63 [==============================] - 12s 186ms/step - loss: 0.3104 - accurac
y: 0.9940 - val_loss: 1.9006 - val_accuracy: 0.9770
Epoch 38/100
63/63 [==============================] - 12s 184ms/step - loss: 0.0460 - accurac
y: 0.9980 - val_loss: 1.8716 - val_accuracy: 0.9770
Epoch 39/100
63/63 [==============================] - 13s 205ms/step - loss: 0.0648 - accurac
y: 0.9985 - val_loss: 2.1534 - val_accuracy: 0.9750
Epoch 40/100
63/63 [==============================] - 11s 179ms/step - loss: 0.1360 - accurac
y: 0.9965 - val_loss: 2.1076 - val_accuracy: 0.9770
Epoch 41/100
63/63 [==============================] - 12s 183ms/step - loss: 0.0937 - accurac
y: 0.9970 - val_loss: 2.1836 - val_accuracy: 0.9800
Epoch 42/100
63/63 [==============================] - 13s 203ms/step - loss: 1.3441e-05 - acc
uracy: 1.0000 - val_loss: 2.1650 - val_accuracy: 0.9780
Epoch 43/100
63/63 [==============================] - 12s 181ms/step - loss: 0.1693 - accurac
y: 0.9970 - val_loss: 2.2611 - val_accuracy: 0.9760
Epoch 44/100
63/63 [==============================] - 12s 187ms/step - loss: 0.0344 - accurac
y: 0.9970 - val_loss: 2.2367 - val_accuracy: 0.9780
Epoch 45/100
63/63 [==============================] - 12s 185ms/step - loss: 0.0890 - accurac
y: 0.9975 - val_loss: 2.3959 - val_accuracy: 0.9770
Epoch 46/100
63/63 [==============================] - 12s 186ms/step - loss: 0.2240 - accurac
y: 0.9960 - val_loss: 2.1349 - val_accuracy: 0.9790
Epoch 47/100
63/63 [==============================] - 12s 181ms/step - loss: 0.0528 - accurac
y: 0.9975 - val_loss: 2.3686 - val_accuracy: 0.9760
Epoch 48/100
63/63 [==============================] - 11s 176ms/step - loss: 0.1274 - accurac
y: 0.9975 - val_loss: 2.2909 - val_accuracy: 0.9750
Epoch 49/100
63/63 [==============================] - 12s 181ms/step - loss: 0.1891 - accurac
y: 0.9970 - val_loss: 2.3322 - val_accuracy: 0.9750
Epoch 50/100
63/63 [==============================] - 12s 187ms/step - loss: 0.0586 - accurac
y: 0.9975 - val_loss: 2.7040 - val_accuracy: 0.9770
Epoch 51/100
63/63 [==============================] - 13s 205ms/step - loss: 0.0580 - accurac
y: 0.9970 - val_loss: 2.3466 - val_accuracy: 0.9780
Epoch 52/100
63/63 [==============================] - 12s 184ms/step - loss: 0.0182 - accurac
y: 0.9985 - val_loss: 2.1442 - val_accuracy: 0.9790
Epoch 53/100
63/63 [==============================] - 12s 183ms/step - loss: 0.0827 - accurac
y: 0.9975 - val_loss: 1.9752 - val_accuracy: 0.9830
Epoch 54/100
63/63 [==============================] - 13s 203ms/step - loss: 0.0543 - accurac
```

```
y: 0.9985 - val_loss: 2.2030 - val_accuracy: 0.9780
Epoch 55/100
63/63 [==============================] - 13s 201ms/step - loss: 0.0590 - accurac
y: 0.9980 - val_loss: 1.7440 - val_accuracy: 0.9840
Epoch 56/100
63/63 [==============================] - 11s 175ms/step - loss: 0.1957 - accurac
y: 0.9970 - val_loss: 2.8776 - val_accuracy: 0.9750
Epoch 57/100
63/63 [==============================] - 12s 180ms/step - loss: 0.0655 - accurac
y: 0.9980 - val_loss: 1.9859 - val_accuracy: 0.9770
Epoch 58/100
63/63 [==============================] - 12s 181ms/step - loss: 0.2062 - accurac
y: 0.9955 - val_loss: 1.7528 - val_accuracy: 0.9770
Epoch 59/100
63/63 [==============================] - 12s 180ms/step - loss: 0.1107 - accurac
y: 0.9970 - val_loss: 1.7113 - val_accuracy: 0.9800
Epoch 60/100
63/63 [==============================] - 12s 184ms/step - loss: 0.0810 - accurac
y: 0.9980 - val_loss: 1.9442 - val_accuracy: 0.9810
Epoch 61/100
63/63 [==============================] - 11s 176ms/step - loss: 0.0458 - accurac
y: 0.9980 - val_loss: 2.1636 - val_accuracy: 0.9780
Epoch 62/100
63/63 [==============================] - 12s 181ms/step - loss: 0.0360 - accurac
y: 0.9985 - val_loss: 2.1349 - val_accuracy: 0.9810
Epoch 63/100
63/63 [==============================] - 13s 202ms/step - loss: 0.0772 - accurac
y: 0.9980 - val_loss: 2.2333 - val_accuracy: 0.9800
Epoch 64/100
63/63 [==============================] - 12s 181ms/step - loss: 0.1846 - accurac
y: 0.9980 - val_loss: 2.2409 - val_accuracy: 0.9780
Epoch 65/100
63/63 [==============================] - 13s 202ms/step - loss: 0.1737 - accurac
y: 0.9975 - val_loss: 2.0886 - val_accuracy: 0.9810
Epoch 66/100
63/63 [==============================] - 12s 181ms/step - loss: 0.0172 - accurac
y: 0.9990 - val_loss: 2.1310 - val_accuracy: 0.9800
Epoch 67/100
63/63 [==============================] - 11s 176ms/step - loss: 1.9351e-07 - acc
uracy: 1.0000 - val_loss: 2.1046 - val_accuracy: 0.9800
Epoch 68/100
63/63 [==============================] - 12s 180ms/step - loss: 0.1203 - accurac
y: 0.9975 - val_loss: 2.1942 - val_accuracy: 0.9770
Epoch 69/100
63/63 [==============================] - 11s 175ms/step - loss: 0.0379 - accurac
y: 0.9995 - val_loss: 2.2327 - val_accuracy: 0.9790
Epoch 70/100
63/63 [==============================] - 12s 182ms/step - loss: 0.0693 - accurac
y: 0.9980 - val_loss: 2.0038 - val_accuracy: 0.9810
Epoch 71/100
63/63 [==============================] - 12s 186ms/step - loss: 0.0394 - accurac
y: 0.9980 - val_loss: 1.9571 - val_accuracy: 0.9780
Epoch 72/100
63/63 [==============================] - 11s 180ms/step - loss: 0.0451 - accurac
y: 0.9970 - val_loss: 2.0473 - val_accuracy: 0.9830
Epoch 73/100
63/63 [==============================] - 12s 184ms/step - loss: 0.1486 - accurac
y: 0.9980 - val_loss: 2.0558 - val_accuracy: 0.9810
Epoch 74/100
63/63 [==============================] - 11s 178ms/step - loss: 1.1523e-09 - acc
uracy: 1.0000 - val_loss: 2.0557 - val_accuracy: 0.9810
Epoch 75/100
63/63 [==============================] - 12s 183ms/step - loss: 0.1281 - accurac
y: 0.9975 - val_loss: 1.9867 - val_accuracy: 0.9810
Epoch 76/100
```

```
63/63 [==============================] - 13s 203ms/step - loss: 0.0765 - accurac
y: 0.9970 - val_loss: 2.1310 - val_accuracy: 0.9790
Epoch 77/100
63/63 [==============================] - 12s 179ms/step - loss: 0.0759 - accurac
y: 0.9970 - val_loss: 2.2644 - val_accuracy: 0.9780
Epoch 78/100
63/63 [==============================] - 12s 180ms/step - loss: 0.1315 - accurac
y: 0.9985 - val_loss: 2.3296 - val_accuracy: 0.9780
Epoch 79/100
63/63 [==============================] - 12s 184ms/step - loss: 0.0919 - accurac
y: 0.9985 - val_loss: 2.2189 - val_accuracy: 0.9810
Epoch 80/100
63/63 [==============================] - 11s 179ms/step - loss: 0.0030 - accurac
y: 0.9985 - val_loss: 2.4759 - val_accuracy: 0.9770
Epoch 81/100
63/63 [==============================] - 12s 181ms/step - loss: 0.1556 - accurac
y: 0.9985 - val_loss: 1.9301 - val_accuracy: 0.9840
Epoch 82/100
63/63 [==============================] - 12s 188ms/step - loss: 0.0652 - accurac
y: 0.9985 - val_loss: 2.7076 - val_accuracy: 0.9770
Epoch 83/100
63/63 [==============================] - 12s 180ms/step - loss: 0.0067 - accurac
y: 0.9995 - val_loss: 2.0469 - val_accuracy: 0.9800
Epoch 84/100
63/63 [==============================] - 12s 186ms/step - loss: 0.0487 - accurac
y: 0.9970 - val_loss: 2.6656 - val_accuracy: 0.9790
Epoch 85/100
63/63 [==============================] - 13s 204ms/step - loss: 0.0984 - accurac
y: 0.9965 - val_loss: 2.3096 - val_accuracy: 0.9800
Epoch 86/100
63/63 [==============================] - 12s 179ms/step - loss: 0.0178 - accurac
y: 0.9995 - val_loss: 3.6648 - val_accuracy: 0.9720
Epoch 87/100
63/63 [==============================] - 12s 183ms/step - loss: 0.0221 - accurac
y: 0.9985 - val_loss: 1.9839 - val_accuracy: 0.9800
Epoch 88/100
63/63 [==============================] - 12s 179ms/step - loss: 0.0349 - accurac
y: 0.9970 - val_loss: 2.0825 - val_accuracy: 0.9810
Epoch 89/100
63/63 [==============================] - 12s 183ms/step - loss: 0.0917 - accurac
y: 0.9980 - val_loss: 1.8227 - val_accuracy: 0.9830
Epoch 90/100
63/63 [==============================] - 12s 178ms/step - loss: 0.0887 - accurac
y: 0.9985 - val_loss: 1.9118 - val_accuracy: 0.9800
Epoch 91/100
63/63 [==============================] - 12s 183ms/step - loss: 0.1035 - accurac
y: 0.9965 - val_loss: 2.2025 - val_accuracy: 0.9790
Epoch 92/100
63/63 [==============================] - 12s 185ms/step - loss: 0.0273 - accurac
y: 0.9995 - val_loss: 2.0092 - val_accuracy: 0.9780
Epoch 93/100
63/63 [==============================] - 12s 181ms/step - loss: 0.0102 - accurac
y: 0.9990 - val_loss: 1.8527 - val_accuracy: 0.9790
Epoch 94/100
63/63 [==============================] - 12s 185ms/step - loss: 0.0389 - accurac
y: 0.9975 - val_loss: 1.7426 - val_accuracy: 0.9800
Epoch 95/100
63/63 [==============================] - 13s 203ms/step - loss: 0.0069 - accurac
y: 0.9995 - val_loss: 1.7382 - val_accuracy: 0.9830
Epoch 96/100
63/63 [==============================] - 12s 183ms/step - loss: 2.4441e-07 - acc
uracy: 1.0000 - val_loss: 1.7318 - val_accuracy: 0.9830
Epoch 97/100
63/63 [==============================] - 11s 178ms/step - loss: 0.0362 - accurac
y: 0.9985 - val_loss: 1.9271 - val_accuracy: 0.9810
```

```
Epoch 98/100
63/63 [==============================] - 13s 193ms/step - loss: 0.0969 - accurac
y: 0.9980 - val_loss: 1.6749 - val_accuracy: 0.9840
Epoch 99/100
63/63 [==============================] - 12s 189ms/step - loss: 0.0127 - accurac
y: 0.9990 - val_loss: 2.1938 - val_accuracy: 0.9780
Epoch 100/100
63/63 [==============================] - 13s 205ms/step - loss: 0.0194 - accurac
y: 0.9985 - val_loss: 1.7454 - val_accuracy: 0.9800
```

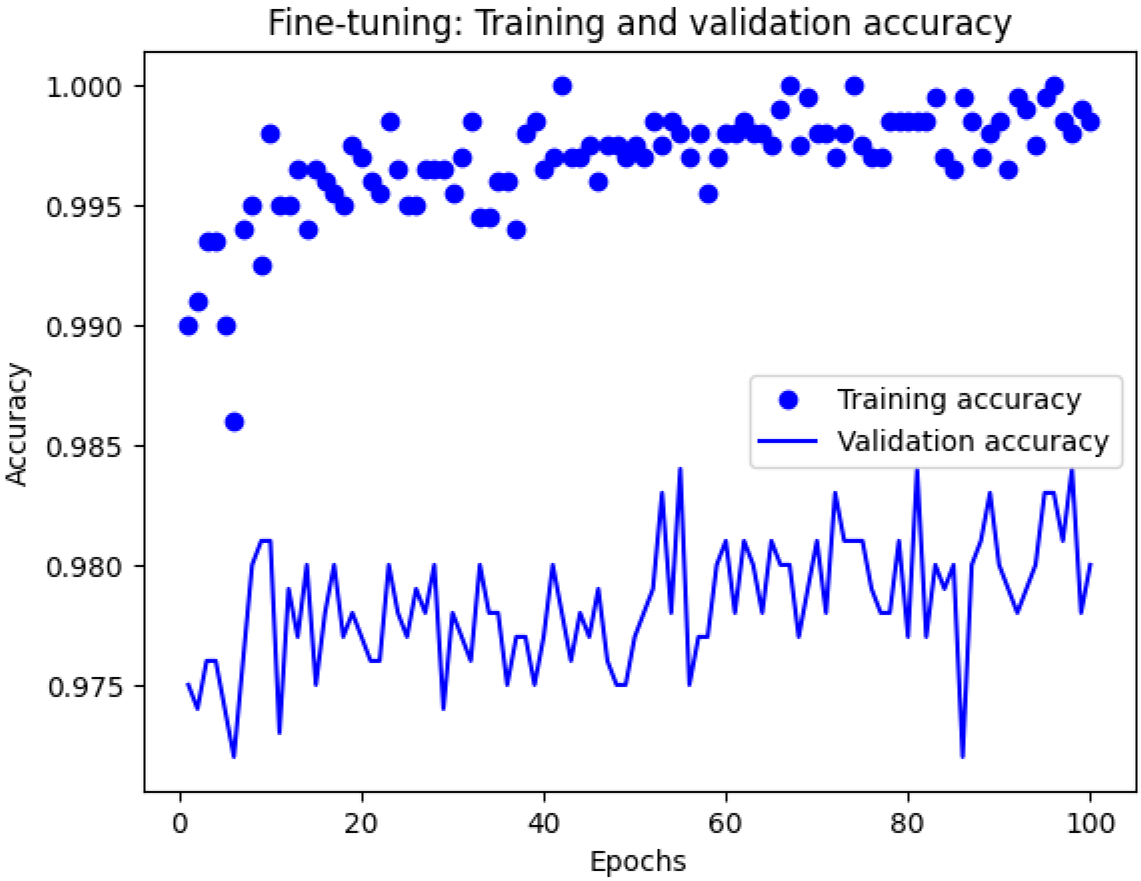
Curves of loss and accuracy during training

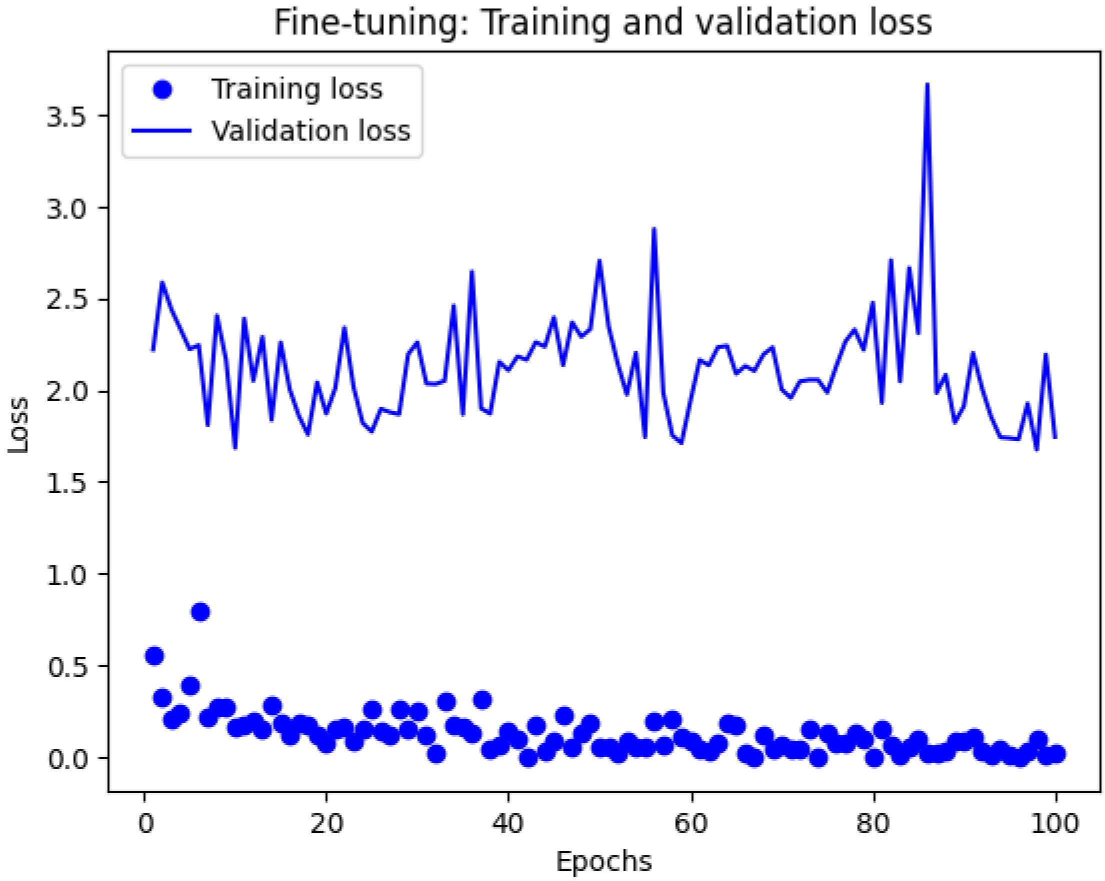
In [39]:

```
accur_tune = historytuning.history["accuracy"]
val_accur_tune = historytuning.history["val_accuracy"]
loss_tune = historytuning.history["loss"]
val_loss_tune = historytuning.history["val_loss"]
epochs = range(1, len(accur_tune) + 1)

plt.plot(epochs, accur_tune, "bo", label="Training accuracy")
plt.plot(epochs, val_accur_tune, "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_tune, "bo", label="Training loss")
plt.plot(epochs, val_loss_tune, "b", label="Validation loss")
plt.title("Fine-tuning: Training and validation loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```





Test Accuracy of model

In [40]:

```
tesaccfi = keras.models.load_model(
"fine_tuning.keras",safe_mode=False)
test_loss, test_acc = tesaccfi.evaluate(test_data)
print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [==============================] - 4s 96ms/step - loss: 2.1343 - accuracy:
0.9820
Test accuracy: 0.982
```

In [ ]: