

HW2

October 13, 2023

```
[ ]: # PART1

[ ]: # import lib

[2]: import tensorflow as tf
import os
import shutil
import pathlib

[3]: import matplotlib.pyplot as plt

[ ]: # setup the path and train validation test segmentation

[10]: original_dir = pathlib.Path('C:\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\train')
new_base_dir = pathlib.Path('C:
↪\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\cats_vs_dogs_hw2')

def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = new_base_dir / subset_name / category
        os.makedirs(dir)
        fnames = [f"{category}.{i}.jpg"
                    for i in range(start_index, end_index)]
        for fname in fnames:
            shutil.copyfile(src=original_dir / fname,
                            dst=dir / fname)
make_subset("train", start_index=0, end_index=1000)          ###1000  ↵
↪###      1000 2000
make_subset("validation", start_index=1000, end_index=1500)  ###500
make_subset("test", start_index=1500, end_index=2000)        ###500 TEST

[4]: from tensorflow import keras
from tensorflow.keras import layers

[ ]: # Defining a new convnet that includes image augmentation and dropout

[ ]: # define datasets
```

```
[12]: from tensorflow.keras.utils import image_dataset_from_directory
```

```
train_dataset = image_dataset_from_directory(
    new_base_dir / "train",
    image_size=(180, 180),
    batch_size=32)
validation_dataset = image_dataset_from_directory(
    new_base_dir / "validation",
    image_size=(180, 180),
    batch_size=32)
test_dataset = image_dataset_from_directory(
    new_base_dir / "test",
    image_size=(180, 180),
    batch_size=32)
```

Found 2000 files belonging to 2 classes.

Found 1000 files belonging to 2 classes.

Found 1000 files belonging to 2 classes.

```
[13]: import numpy as np
import tensorflow as tf
random_numbers = np.random.normal(size=(1000, 16))
dataset = tf.data.Dataset.from_tensor_slices(random_numbers)
```

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

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

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

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

```
for data_batch, labels_batch in train_dataset:
    print("data batch shape:", data_batch.shape)
    print("labels batch shape:", labels_batch.shape)
    break
```

(16,)

(16,)

(16,)

```
(32, 16)
(32, 16)
(32, 16)
(4, 4)
(4, 4)
(4, 4)
data batch shape: (32, 180, 180, 3)
labels batch shape: (32,)
```

```
[14]: data_augmentation = keras.Sequential(
      [
          layers.RandomFlip("horizontal"),
          layers.RandomRotation(0.1),
          layers.RandomZoom(0.2),
      ]
  )
```

```
[15]: plt.figure(figsize=(10, 10))
      for images, _ in train_dataset.take(1):
          for i in range(9):
              augmented_images = data_augmentation(images)
              ax = plt.subplot(3, 3, i + 1)
              plt.imshow(augmented_images[0].numpy().astype("uint8"))
              plt.axis("off")
```

```
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
```

registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
 WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
 WARNING:tensorflow:5 out of the last 5 calls to <function pfor.<locals>.f at

0x00000240BC29A5E0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:6 out of the last 6 calls to <function pfor.<locals>.f at 0x00000240BC29A430> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there

```

is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while loop for converting ImageProjectiveTransformV3

```

```

cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while loop for converting StatelessRandomUniformV2

```

cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.



```
[16]: # model buildup
inputs = keras.Input(shape=(180, 180, 3))
x = data_augmentation(inputs)
x = layers.Rescaling(1./255)(x)
x = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(x)
```



```

x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
x = layers.Flatten()(x)
x = layers.Dropout(0.4)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs=inputs, outputs=outputs)

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

```

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no

registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

```
[18]: # Training the regularized convnet
callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="convnet_from_scratch_with_augmentation_regularization_hw2.
↳keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=150,
    validation_data=validation_dataset,
    callbacks=callbacks)
```

Epoch 1/150

63/63 [=====] - 13s 200ms/step - loss: 0.7052 - accuracy: 0.5005 - val_loss: 0.6919 - val_accuracy: 0.5020

Epoch 2/150

63/63 [=====] - 9s 140ms/step - loss: 0.6949 - accuracy: 0.5525 - val_loss: 0.6885 - val_accuracy: 0.5350

Epoch 3/150

63/63 [=====] - 9s 137ms/step - loss: 0.6882 - accuracy: 0.5940 - val_loss: 0.6564 - val_accuracy: 0.6090

Epoch 4/150

63/63 [=====] - 9s 139ms/step - loss: 0.6513 - accuracy: 0.6220 - val_loss: 0.6112 - val_accuracy: 0.6660

Epoch 5/150

63/63 [=====] - 9s 139ms/step - loss: 0.6288 - accuracy: 0.6585 - val_loss: 0.6109 - val_accuracy: 0.6480

Epoch 6/150

63/63 [=====] - 9s 139ms/step - loss: 0.6118 - accuracy: 0.6830 - val_loss: 0.5908 - val_accuracy: 0.6790

Epoch 7/150

63/63 [=====] - 9s 135ms/step - loss: 0.5926 - accuracy: 0.6810 - val_loss: 0.6298 - val_accuracy: 0.6560

Epoch 8/150

63/63 [=====] - 9s 137ms/step - loss: 0.6057 - accuracy: 0.6865 - val_loss: 0.6111 - val_accuracy: 0.6620

Epoch 9/150

63/63 [=====] - 9s 138ms/step - loss: 0.5724 - accuracy: 0.7095 - val_loss: 0.5828 - val_accuracy: 0.6810

Epoch 10/150

63/63 [=====] - 9s 139ms/step - loss: 0.5669 -
accuracy: 0.7155 - val_loss: 0.5511 - val_accuracy: 0.7210
Epoch 11/150
63/63 [=====] - 9s 140ms/step - loss: 0.5530 -
accuracy: 0.7050 - val_loss: 0.6653 - val_accuracy: 0.6860
Epoch 12/150
63/63 [=====] - 9s 138ms/step - loss: 0.5411 -
accuracy: 0.7260 - val_loss: 0.6144 - val_accuracy: 0.7050
Epoch 13/150
63/63 [=====] - 8s 134ms/step - loss: 0.5218 -
accuracy: 0.7420 - val_loss: 0.5516 - val_accuracy: 0.7380
Epoch 14/150
63/63 [=====] - 9s 138ms/step - loss: 0.5260 -
accuracy: 0.7375 - val_loss: 0.5770 - val_accuracy: 0.7010
Epoch 15/150
63/63 [=====] - 9s 140ms/step - loss: 0.5110 -
accuracy: 0.7625 - val_loss: 0.5125 - val_accuracy: 0.7720
Epoch 16/150
63/63 [=====] - 8s 134ms/step - loss: 0.4922 -
accuracy: 0.7655 - val_loss: 0.5314 - val_accuracy: 0.7600
Epoch 17/150
63/63 [=====] - 9s 138ms/step - loss: 0.4727 -
accuracy: 0.7730 - val_loss: 0.5228 - val_accuracy: 0.7530
Epoch 18/150
63/63 [=====] - 9s 137ms/step - loss: 0.4773 -
accuracy: 0.7715 - val_loss: 0.8790 - val_accuracy: 0.6640
Epoch 19/150
63/63 [=====] - 9s 142ms/step - loss: 0.4536 -
accuracy: 0.7880 - val_loss: 0.5129 - val_accuracy: 0.7600
Epoch 20/150
63/63 [=====] - 9s 144ms/step - loss: 0.4434 -
accuracy: 0.7940 - val_loss: 0.4945 - val_accuracy: 0.7760
Epoch 21/150
63/63 [=====] - 9s 140ms/step - loss: 0.4362 -
accuracy: 0.7955 - val_loss: 0.5245 - val_accuracy: 0.7610
Epoch 22/150
63/63 [=====] - 9s 140ms/step - loss: 0.4149 -
accuracy: 0.8255 - val_loss: 0.5721 - val_accuracy: 0.7410
Epoch 23/150
63/63 [=====] - 9s 141ms/step - loss: 0.4003 -
accuracy: 0.8220 - val_loss: 0.5206 - val_accuracy: 0.8030
Epoch 24/150
63/63 [=====] - 9s 145ms/step - loss: 0.3899 -
accuracy: 0.8280 - val_loss: 0.5190 - val_accuracy: 0.7950
Epoch 25/150
63/63 [=====] - 9s 140ms/step - loss: 0.3759 -
accuracy: 0.8255 - val_loss: 0.4647 - val_accuracy: 0.7950
Epoch 26/150

63/63 [=====] - 9s 139ms/step - loss: 0.3469 -
accuracy: 0.8495 - val_loss: 0.4997 - val_accuracy: 0.7940
Epoch 27/150
63/63 [=====] - 9s 139ms/step - loss: 0.3628 -
accuracy: 0.8460 - val_loss: 0.6388 - val_accuracy: 0.7670
Epoch 28/150
63/63 [=====] - 9s 139ms/step - loss: 0.3404 -
accuracy: 0.8615 - val_loss: 0.6373 - val_accuracy: 0.7740
Epoch 29/150
63/63 [=====] - 9s 138ms/step - loss: 0.3451 -
accuracy: 0.8530 - val_loss: 0.4990 - val_accuracy: 0.8050
Epoch 30/150
63/63 [=====] - 9s 136ms/step - loss: 0.3256 -
accuracy: 0.8615 - val_loss: 0.7732 - val_accuracy: 0.7860
Epoch 31/150
63/63 [=====] - 9s 138ms/step - loss: 0.3288 -
accuracy: 0.8660 - val_loss: 0.4891 - val_accuracy: 0.7950
Epoch 32/150
63/63 [=====] - 9s 140ms/step - loss: 0.3052 -
accuracy: 0.8675 - val_loss: 0.6160 - val_accuracy: 0.7980
Epoch 33/150
63/63 [=====] - 9s 138ms/step - loss: 0.2889 -
accuracy: 0.8815 - val_loss: 0.5812 - val_accuracy: 0.8140
Epoch 34/150
63/63 [=====] - 9s 141ms/step - loss: 0.2970 -
accuracy: 0.8800 - val_loss: 0.5772 - val_accuracy: 0.8090
Epoch 35/150
63/63 [=====] - 9s 141ms/step - loss: 0.2845 -
accuracy: 0.8780 - val_loss: 0.4859 - val_accuracy: 0.8210
Epoch 36/150
63/63 [=====] - 9s 137ms/step - loss: 0.2726 -
accuracy: 0.8825 - val_loss: 0.5597 - val_accuracy: 0.8210
Epoch 37/150
63/63 [=====] - 9s 139ms/step - loss: 0.2694 -
accuracy: 0.8945 - val_loss: 0.6155 - val_accuracy: 0.7860
Epoch 38/150
63/63 [=====] - 9s 139ms/step - loss: 0.2849 -
accuracy: 0.8805 - val_loss: 0.5743 - val_accuracy: 0.8160
Epoch 39/150
63/63 [=====] - 9s 137ms/step - loss: 0.2526 -
accuracy: 0.8905 - val_loss: 0.5447 - val_accuracy: 0.7930
Epoch 40/150
63/63 [=====] - 9s 135ms/step - loss: 0.2380 -
accuracy: 0.9040 - val_loss: 0.5204 - val_accuracy: 0.8170
Epoch 41/150
63/63 [=====] - 9s 140ms/step - loss: 0.2473 -
accuracy: 0.9065 - val_loss: 0.5272 - val_accuracy: 0.8410
Epoch 42/150

63/63 [=====] - 9s 137ms/step - loss: 0.2265 -
accuracy: 0.9060 - val_loss: 0.7182 - val_accuracy: 0.7900
Epoch 43/150
63/63 [=====] - 9s 141ms/step - loss: 0.2120 -
accuracy: 0.9180 - val_loss: 0.6500 - val_accuracy: 0.8200
Epoch 44/150
63/63 [=====] - 9s 141ms/step - loss: 0.2153 -
accuracy: 0.9215 - val_loss: 0.7917 - val_accuracy: 0.7950
Epoch 45/150
63/63 [=====] - 9s 140ms/step - loss: 0.2007 -
accuracy: 0.9110 - val_loss: 0.5872 - val_accuracy: 0.8230
Epoch 46/150
63/63 [=====] - 9s 142ms/step - loss: 0.1883 -
accuracy: 0.9220 - val_loss: 0.6692 - val_accuracy: 0.7900
Epoch 47/150
63/63 [=====] - 9s 142ms/step - loss: 0.1991 -
accuracy: 0.9220 - val_loss: 0.6406 - val_accuracy: 0.8160
Epoch 48/150
63/63 [=====] - 9s 141ms/step - loss: 0.2082 -
accuracy: 0.9145 - val_loss: 0.5933 - val_accuracy: 0.8030
Epoch 49/150
63/63 [=====] - 9s 140ms/step - loss: 0.1911 -
accuracy: 0.9260 - val_loss: 0.5724 - val_accuracy: 0.8380
Epoch 50/150
63/63 [=====] - 9s 140ms/step - loss: 0.1810 -
accuracy: 0.9230 - val_loss: 0.6121 - val_accuracy: 0.8270
Epoch 51/150
63/63 [=====] - 9s 142ms/step - loss: 0.1760 -
accuracy: 0.9275 - val_loss: 0.7282 - val_accuracy: 0.8140
Epoch 52/150
63/63 [=====] - 9s 144ms/step - loss: 0.1917 -
accuracy: 0.9275 - val_loss: 0.6795 - val_accuracy: 0.7940
Epoch 53/150
63/63 [=====] - 9s 142ms/step - loss: 0.1760 -
accuracy: 0.9290 - val_loss: 0.7158 - val_accuracy: 0.8050
Epoch 54/150
63/63 [=====] - 9s 142ms/step - loss: 0.1870 -
accuracy: 0.9355 - val_loss: 0.9229 - val_accuracy: 0.7880
Epoch 55/150
63/63 [=====] - 9s 142ms/step - loss: 0.1711 -
accuracy: 0.9290 - val_loss: 0.6480 - val_accuracy: 0.8380
Epoch 56/150
63/63 [=====] - 9s 142ms/step - loss: 0.1628 -
accuracy: 0.9370 - val_loss: 0.8357 - val_accuracy: 0.8150
Epoch 57/150
63/63 [=====] - 9s 144ms/step - loss: 0.1733 -
accuracy: 0.9400 - val_loss: 0.7561 - val_accuracy: 0.7990
Epoch 58/150

63/63 [=====] - 9s 143ms/step - loss: 0.1528 -
accuracy: 0.9405 - val_loss: 0.8247 - val_accuracy: 0.8170
Epoch 59/150
63/63 [=====] - 9s 144ms/step - loss: 0.1549 -
accuracy: 0.9380 - val_loss: 0.6286 - val_accuracy: 0.8160
Epoch 60/150
63/63 [=====] - 9s 144ms/step - loss: 0.1520 -
accuracy: 0.9430 - val_loss: 0.6469 - val_accuracy: 0.8330
Epoch 61/150
63/63 [=====] - 9s 147ms/step - loss: 0.1823 -
accuracy: 0.9350 - val_loss: 0.8010 - val_accuracy: 0.8180
Epoch 62/150
63/63 [=====] - 10s 151ms/step - loss: 0.1491 -
accuracy: 0.9450 - val_loss: 0.7691 - val_accuracy: 0.8230
Epoch 63/150
63/63 [=====] - 9s 144ms/step - loss: 0.1564 -
accuracy: 0.9430 - val_loss: 1.5770 - val_accuracy: 0.7400
Epoch 64/150
63/63 [=====] - 9s 143ms/step - loss: 0.1654 -
accuracy: 0.9405 - val_loss: 0.6822 - val_accuracy: 0.8280
Epoch 65/150
63/63 [=====] - 9s 138ms/step - loss: 0.1485 -
accuracy: 0.9515 - val_loss: 0.7889 - val_accuracy: 0.8090
Epoch 66/150
63/63 [=====] - 9s 142ms/step - loss: 0.1502 -
accuracy: 0.9440 - val_loss: 0.6516 - val_accuracy: 0.8180
Epoch 67/150
63/63 [=====] - 9s 139ms/step - loss: 0.1486 -
accuracy: 0.9485 - val_loss: 0.8667 - val_accuracy: 0.8180
Epoch 68/150
63/63 [=====] - 9s 142ms/step - loss: 0.1330 -
accuracy: 0.9480 - val_loss: 0.7437 - val_accuracy: 0.8290
Epoch 69/150
63/63 [=====] - 9s 138ms/step - loss: 0.1491 -
accuracy: 0.9440 - val_loss: 0.7235 - val_accuracy: 0.8270
Epoch 70/150
63/63 [=====] - 9s 137ms/step - loss: 0.1351 -
accuracy: 0.9495 - val_loss: 0.9074 - val_accuracy: 0.8220
Epoch 71/150
63/63 [=====] - 9s 139ms/step - loss: 0.1399 -
accuracy: 0.9495 - val_loss: 1.2545 - val_accuracy: 0.7710
Epoch 72/150
63/63 [=====] - 9s 138ms/step - loss: 0.1496 -
accuracy: 0.9525 - val_loss: 0.6743 - val_accuracy: 0.8410
Epoch 73/150
63/63 [=====] - 9s 136ms/step - loss: 0.1093 -
accuracy: 0.9600 - val_loss: 0.8690 - val_accuracy: 0.8490
Epoch 74/150

63/63 [=====] - 9s 139ms/step - loss: 0.1334 -
accuracy: 0.9560 - val_loss: 0.7178 - val_accuracy: 0.8380
Epoch 75/150
63/63 [=====] - 9s 137ms/step - loss: 0.1420 -
accuracy: 0.9535 - val_loss: 0.7657 - val_accuracy: 0.8250
Epoch 76/150
63/63 [=====] - 9s 139ms/step - loss: 0.1430 -
accuracy: 0.9545 - val_loss: 1.3529 - val_accuracy: 0.8030
Epoch 77/150
63/63 [=====] - 9s 140ms/step - loss: 0.1382 -
accuracy: 0.9505 - val_loss: 0.7825 - val_accuracy: 0.8320
Epoch 78/150
63/63 [=====] - 9s 137ms/step - loss: 0.1325 -
accuracy: 0.9550 - val_loss: 0.7365 - val_accuracy: 0.8360
Epoch 79/150
63/63 [=====] - 9s 139ms/step - loss: 0.1402 -
accuracy: 0.9480 - val_loss: 0.7766 - val_accuracy: 0.8190
Epoch 80/150
63/63 [=====] - 9s 140ms/step - loss: 0.1332 -
accuracy: 0.9535 - val_loss: 0.7476 - val_accuracy: 0.8210
Epoch 81/150
63/63 [=====] - 9s 144ms/step - loss: 0.1184 -
accuracy: 0.9575 - val_loss: 0.9348 - val_accuracy: 0.8170
Epoch 82/150
63/63 [=====] - 9s 142ms/step - loss: 0.1211 -
accuracy: 0.9520 - val_loss: 0.7366 - val_accuracy: 0.8330
Epoch 83/150
63/63 [=====] - 9s 142ms/step - loss: 0.1227 -
accuracy: 0.9575 - val_loss: 0.7608 - val_accuracy: 0.8350
Epoch 84/150
63/63 [=====] - 9s 139ms/step - loss: 0.1257 -
accuracy: 0.9640 - val_loss: 1.1204 - val_accuracy: 0.7840
Epoch 85/150
63/63 [=====] - 9s 141ms/step - loss: 0.1322 -
accuracy: 0.9540 - val_loss: 1.0605 - val_accuracy: 0.8140
Epoch 86/150
63/63 [=====] - 9s 142ms/step - loss: 0.1260 -
accuracy: 0.9655 - val_loss: 0.9556 - val_accuracy: 0.8280
Epoch 87/150
63/63 [=====] - 9s 142ms/step - loss: 0.1474 -
accuracy: 0.9540 - val_loss: 0.8023 - val_accuracy: 0.8220
Epoch 88/150
63/63 [=====] - 9s 144ms/step - loss: 0.1174 -
accuracy: 0.9630 - val_loss: 1.0391 - val_accuracy: 0.8270
Epoch 89/150
63/63 [=====] - 9s 141ms/step - loss: 0.1476 -
accuracy: 0.9500 - val_loss: 0.8349 - val_accuracy: 0.8340
Epoch 90/150

63/63 [=====] - 9s 142ms/step - loss: 0.1202 -
accuracy: 0.9620 - val_loss: 0.7866 - val_accuracy: 0.8240
Epoch 91/150
63/63 [=====] - 9s 142ms/step - loss: 0.0994 -
accuracy: 0.9695 - val_loss: 1.1657 - val_accuracy: 0.8250
Epoch 92/150
63/63 [=====] - 9s 138ms/step - loss: 0.1140 -
accuracy: 0.9595 - val_loss: 1.1493 - val_accuracy: 0.8420
Epoch 93/150
63/63 [=====] - 9s 142ms/step - loss: 0.1376 -
accuracy: 0.9550 - val_loss: 0.9933 - val_accuracy: 0.8230
Epoch 94/150
63/63 [=====] - 9s 137ms/step - loss: 0.1186 -
accuracy: 0.9610 - val_loss: 1.0491 - val_accuracy: 0.8280
Epoch 95/150
63/63 [=====] - 10s 153ms/step - loss: 0.1181 -
accuracy: 0.9620 - val_loss: 1.0390 - val_accuracy: 0.8360
Epoch 96/150
63/63 [=====] - 9s 138ms/step - loss: 0.1214 -
accuracy: 0.9615 - val_loss: 0.9854 - val_accuracy: 0.8170
Epoch 97/150
63/63 [=====] - 9s 142ms/step - loss: 0.1160 -
accuracy: 0.9670 - val_loss: 0.9513 - val_accuracy: 0.8270
Epoch 98/150
63/63 [=====] - 9s 143ms/step - loss: 0.1161 -
accuracy: 0.9605 - val_loss: 1.0714 - val_accuracy: 0.8190
Epoch 99/150
63/63 [=====] - 9s 143ms/step - loss: 0.1375 -
accuracy: 0.9595 - val_loss: 0.9213 - val_accuracy: 0.8290
Epoch 100/150
63/63 [=====] - 9s 146ms/step - loss: 0.1086 -
accuracy: 0.9710 - val_loss: 0.9142 - val_accuracy: 0.8520
Epoch 101/150
63/63 [=====] - 9s 144ms/step - loss: 0.1167 -
accuracy: 0.9640 - val_loss: 0.8603 - val_accuracy: 0.8520
Epoch 102/150
63/63 [=====] - 9s 146ms/step - loss: 0.1028 -
accuracy: 0.9655 - val_loss: 1.3474 - val_accuracy: 0.8010
Epoch 103/150
63/63 [=====] - 9s 147ms/step - loss: 0.1274 -
accuracy: 0.9600 - val_loss: 0.9081 - val_accuracy: 0.8340
Epoch 104/150
63/63 [=====] - 9s 148ms/step - loss: 0.0992 -
accuracy: 0.9665 - val_loss: 1.0052 - val_accuracy: 0.8240
Epoch 105/150
63/63 [=====] - 9s 146ms/step - loss: 0.1236 -
accuracy: 0.9645 - val_loss: 0.8650 - val_accuracy: 0.8490
Epoch 106/150

63/63 [=====] - 9s 145ms/step - loss: 0.1059 -
accuracy: 0.9640 - val_loss: 1.0165 - val_accuracy: 0.8310
Epoch 107/150
63/63 [=====] - 9s 140ms/step - loss: 0.1186 -
accuracy: 0.9640 - val_loss: 1.2783 - val_accuracy: 0.8290
Epoch 108/150
63/63 [=====] - 9s 137ms/step - loss: 0.1069 -
accuracy: 0.9600 - val_loss: 1.3549 - val_accuracy: 0.8350
Epoch 109/150
63/63 [=====] - 9s 141ms/step - loss: 0.1268 -
accuracy: 0.9620 - val_loss: 0.8693 - val_accuracy: 0.8490
Epoch 110/150
63/63 [=====] - 9s 135ms/step - loss: 0.1306 -
accuracy: 0.9595 - val_loss: 1.0174 - val_accuracy: 0.8310
Epoch 111/150
63/63 [=====] - 9s 138ms/step - loss: 0.1071 -
accuracy: 0.9680 - val_loss: 1.1976 - val_accuracy: 0.8320
Epoch 112/150
63/63 [=====] - 9s 136ms/step - loss: 0.0960 -
accuracy: 0.9675 - val_loss: 0.9389 - val_accuracy: 0.8460
Epoch 113/150
63/63 [=====] - 9s 135ms/step - loss: 0.1298 -
accuracy: 0.9595 - val_loss: 1.0293 - val_accuracy: 0.8150
Epoch 114/150
63/63 [=====] - 9s 137ms/step - loss: 0.1201 -
accuracy: 0.9650 - val_loss: 0.9403 - val_accuracy: 0.8380
Epoch 115/150
63/63 [=====] - 9s 138ms/step - loss: 0.1202 -
accuracy: 0.9655 - val_loss: 1.0147 - val_accuracy: 0.8240
Epoch 116/150
63/63 [=====] - 9s 138ms/step - loss: 0.1174 -
accuracy: 0.9670 - val_loss: 0.7992 - val_accuracy: 0.8440
Epoch 117/150
63/63 [=====] - 9s 139ms/step - loss: 0.1187 -
accuracy: 0.9645 - val_loss: 1.0747 - val_accuracy: 0.8270
Epoch 118/150
63/63 [=====] - 9s 139ms/step - loss: 0.1246 -
accuracy: 0.9675 - val_loss: 1.0751 - val_accuracy: 0.8330
Epoch 119/150
63/63 [=====] - 9s 137ms/step - loss: 0.1299 -
accuracy: 0.9665 - val_loss: 1.1499 - val_accuracy: 0.8330
Epoch 120/150
63/63 [=====] - 9s 138ms/step - loss: 0.1227 -
accuracy: 0.9670 - val_loss: 1.3367 - val_accuracy: 0.8130
Epoch 121/150
63/63 [=====] - 9s 144ms/step - loss: 0.1109 -
accuracy: 0.9670 - val_loss: 1.3568 - val_accuracy: 0.8270
Epoch 122/150

63/63 [=====] - 9s 145ms/step - loss: 0.1149 -
accuracy: 0.9625 - val_loss: 1.2270 - val_accuracy: 0.8000
Epoch 123/150
63/63 [=====] - 10s 158ms/step - loss: 0.1424 -
accuracy: 0.9640 - val_loss: 0.9302 - val_accuracy: 0.8350
Epoch 124/150
63/63 [=====] - 9s 139ms/step - loss: 0.1121 -
accuracy: 0.9655 - val_loss: 1.1295 - val_accuracy: 0.8290
Epoch 125/150
63/63 [=====] - 9s 139ms/step - loss: 0.1171 -
accuracy: 0.9685 - val_loss: 0.8872 - val_accuracy: 0.8490
Epoch 126/150
63/63 [=====] - 9s 138ms/step - loss: 0.0957 -
accuracy: 0.9715 - val_loss: 1.3126 - val_accuracy: 0.8280
Epoch 127/150
63/63 [=====] - 9s 141ms/step - loss: 0.1308 -
accuracy: 0.9605 - val_loss: 1.3976 - val_accuracy: 0.8400
Epoch 128/150
63/63 [=====] - 9s 138ms/step - loss: 0.1183 -
accuracy: 0.9725 - val_loss: 1.2293 - val_accuracy: 0.8040
Epoch 129/150
63/63 [=====] - 9s 138ms/step - loss: 0.0889 -
accuracy: 0.9730 - val_loss: 1.2686 - val_accuracy: 0.8390
Epoch 130/150
63/63 [=====] - 9s 139ms/step - loss: 0.1098 -
accuracy: 0.9730 - val_loss: 1.2013 - val_accuracy: 0.8480
Epoch 131/150
63/63 [=====] - 9s 138ms/step - loss: 0.0972 -
accuracy: 0.9705 - val_loss: 1.3291 - val_accuracy: 0.8150
Epoch 132/150
63/63 [=====] - 9s 142ms/step - loss: 0.1523 -
accuracy: 0.9635 - val_loss: 1.1667 - val_accuracy: 0.8220
Epoch 133/150
63/63 [=====] - 9s 137ms/step - loss: 0.1060 -
accuracy: 0.9650 - val_loss: 1.3240 - val_accuracy: 0.8290
Epoch 134/150
63/63 [=====] - 9s 137ms/step - loss: 0.1544 -
accuracy: 0.9560 - val_loss: 0.8965 - val_accuracy: 0.8360
Epoch 135/150
63/63 [=====] - 9s 136ms/step - loss: 0.1071 -
accuracy: 0.9680 - val_loss: 1.0021 - val_accuracy: 0.8540
Epoch 136/150
63/63 [=====] - 9s 137ms/step - loss: 0.0830 -
accuracy: 0.9765 - val_loss: 1.1884 - val_accuracy: 0.8380
Epoch 137/150
63/63 [=====] - 9s 137ms/step - loss: 0.1317 -
accuracy: 0.9655 - val_loss: 1.1081 - val_accuracy: 0.8340
Epoch 138/150

```

63/63 [=====] - 9s 139ms/step - loss: 0.1076 -
accuracy: 0.9690 - val_loss: 1.4140 - val_accuracy: 0.8320
Epoch 139/150
63/63 [=====] - 9s 138ms/step - loss: 0.1163 -
accuracy: 0.9710 - val_loss: 1.2294 - val_accuracy: 0.8370
Epoch 140/150
63/63 [=====] - 9s 138ms/step - loss: 0.1162 -
accuracy: 0.9695 - val_loss: 1.2345 - val_accuracy: 0.8340
Epoch 141/150
63/63 [=====] - 9s 139ms/step - loss: 0.1000 -
accuracy: 0.9700 - val_loss: 1.2069 - val_accuracy: 0.8560
Epoch 142/150
63/63 [=====] - 9s 141ms/step - loss: 0.1248 -
accuracy: 0.9705 - val_loss: 1.4421 - val_accuracy: 0.8440
Epoch 143/150
63/63 [=====] - 9s 138ms/step - loss: 0.1123 -
accuracy: 0.9680 - val_loss: 1.5053 - val_accuracy: 0.8320
Epoch 144/150
63/63 [=====] - 9s 140ms/step - loss: 0.1054 -
accuracy: 0.9710 - val_loss: 1.7306 - val_accuracy: 0.8310
Epoch 145/150
63/63 [=====] - 9s 140ms/step - loss: 0.1508 -
accuracy: 0.9670 - val_loss: 1.9713 - val_accuracy: 0.8270
Epoch 146/150
63/63 [=====] - 9s 137ms/step - loss: 0.1296 -
accuracy: 0.9615 - val_loss: 1.4302 - val_accuracy: 0.8220
Epoch 147/150
63/63 [=====] - 9s 139ms/step - loss: 0.1328 -
accuracy: 0.9735 - val_loss: 1.2410 - val_accuracy: 0.8620
Epoch 148/150
63/63 [=====] - 9s 137ms/step - loss: 0.1370 -
accuracy: 0.9660 - val_loss: 1.2451 - val_accuracy: 0.8290
Epoch 149/150
63/63 [=====] - 9s 138ms/step - loss: 0.1304 -
accuracy: 0.9630 - val_loss: 1.0710 - val_accuracy: 0.8330
Epoch 150/150
63/63 [=====] - 9s 138ms/step - loss: 0.0841 -
accuracy: 0.9705 - val_loss: 1.9029 - val_accuracy: 0.8210

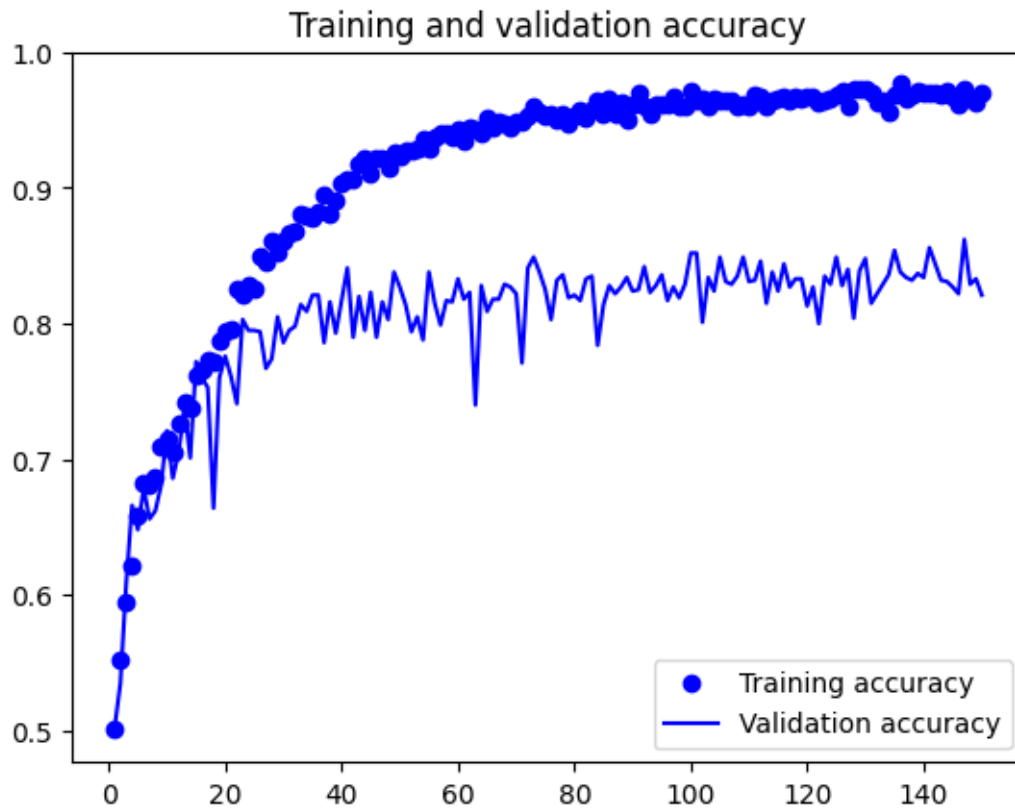
```

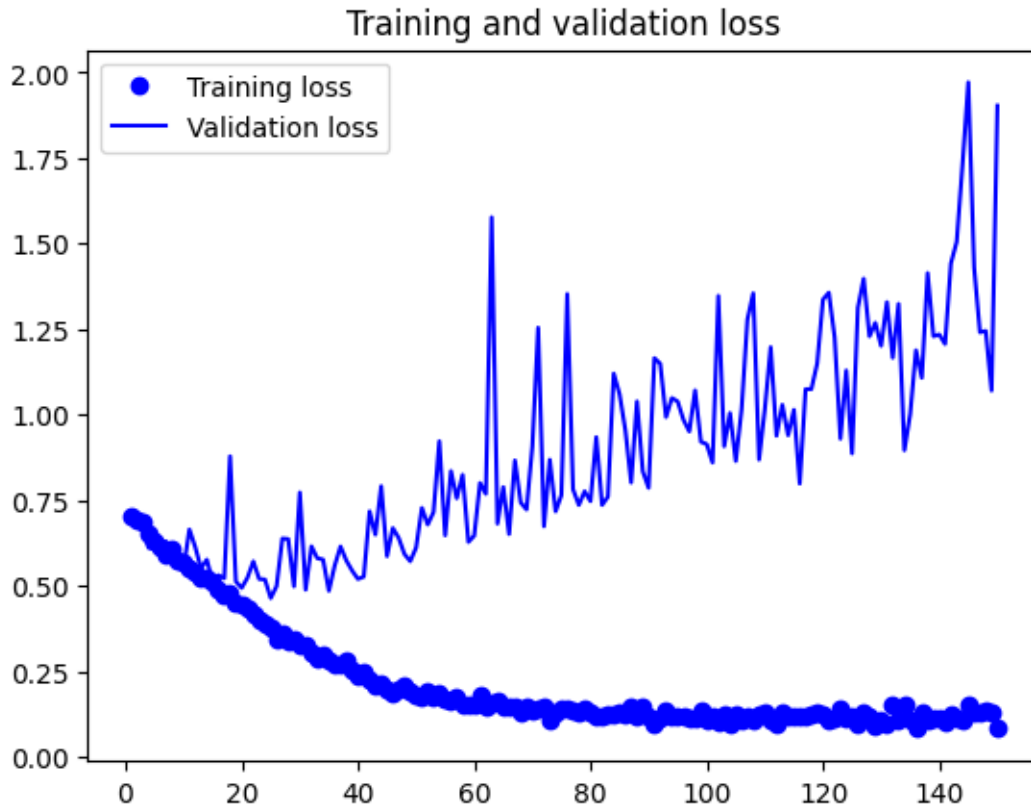
```

[19]: # plot the performance
import matplotlib.pyplot as plt
accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
loss = history.history["loss"]
val_loss = history.history["val_loss"]
epochs = range(1, len(accuracy) + 1)
plt.plot(epochs, accuracy, "bo", label="Training accuracy")

```

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





```
[20]: # Evaluating the model on the test set
```

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

```
32/32 [=====] - 1s 25ms/step - loss: 0.5208 - accuracy: 0.7720
```

```
Test accuracy: 0.772
```

```
[ ]: # PART2
```

```
    # INCREASE TRAINING SIZE TO 4000
```

```
[7]: original_dir = pathlib.Path('C:\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\train')
new_base_dir = pathlib.Path('C:
    ↪\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\cats_vs_dogs_hw2_BIG')
```

```
def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
```

```

    dir = new_base_dir / subset_name / category
    os.makedirs(dir)
    fnames = [f"{category}.{i}.jpg"
               for i in range(start_index, end_index)]
    for fname in fnames:
        shutil.copyfile(src=original_dir / fname,
                        dst=dir / fname)
make_subset("train", start_index=0, end_index=4000)          ###4000
↪###      4000 8000
make_subset("validation", start_index=4000, end_index=4500)  ###500
make_subset("test", start_index=4500, end_index=5000)        ###500 TEST

```

```

[8]: from tensorflow import keras
     from tensorflow.keras import layers

```

```

[9]: from tensorflow.keras.utils import image_dataset_from_directory

```

```

train_dataset = image_dataset_from_directory(
    new_base_dir / "train",
    image_size=(180, 180),
    batch_size=32)
validation_dataset = image_dataset_from_directory(
    new_base_dir / "validation",
    image_size=(180, 180),
    batch_size=32)
test_dataset = image_dataset_from_directory(
    new_base_dir / "test",
    image_size=(180, 180),
    batch_size=32)

```

Found 8000 files belonging to 2 classes.
Found 1000 files belonging to 2 classes.
Found 1000 files belonging to 2 classes.

```

[10]: import numpy as np
      import tensorflow as tf
      random_numbers = np.random.normal(size=(1000, 16))
      dataset = tf.data.Dataset.from_tensor_slices(random_numbers)

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

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

```

```

    if i >= 2:
        break
reshaped_dataset = dataset.map(lambda x: tf.reshape(x, (4, 4)))
for i, element in enumerate(reshaped_dataset):
    print(element.shape)
    if i >= 2:
        break
#Displaying the shapes of the data and labels yielded by the Dataset
for data_batch, labels_batch in train_dataset:
    print("data batch shape:", data_batch.shape)
    print("labels batch shape:", labels_batch.shape)
    break

```

```

(16,)
(16,)
(16,)
(32, 16)
(32, 16)
(32, 16)
(4, 4)
(4, 4)
(4, 4)
data batch shape: (32, 180, 180, 3)
labels batch shape: (32,)

```

```

[11]: data_augmentation = keras.Sequential(
    [
        layers.RandomFlip("horizontal"),
        layers.RandomRotation(0.15),
        layers.RandomZoom(0.25),
    ]
)

```

```

[12]: plt.figure(figsize=(10, 10))
for images, _ in train_dataset.take(1):
    for i in range(9):
        augmented_images = data_augmentation(images)
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(augmented_images[0].numpy().astype("uint8"))
        plt.axis("off")

```

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2

cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no

registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:5 out of the last 5 calls to <function pfor.<locals>.f at 0x0000014AD8000F70> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:6 out of the last 6 calls to <function pfor.<locals>.f at 0x0000014AD8000D30> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

```

registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while loop for converting Bitcast cause there is no

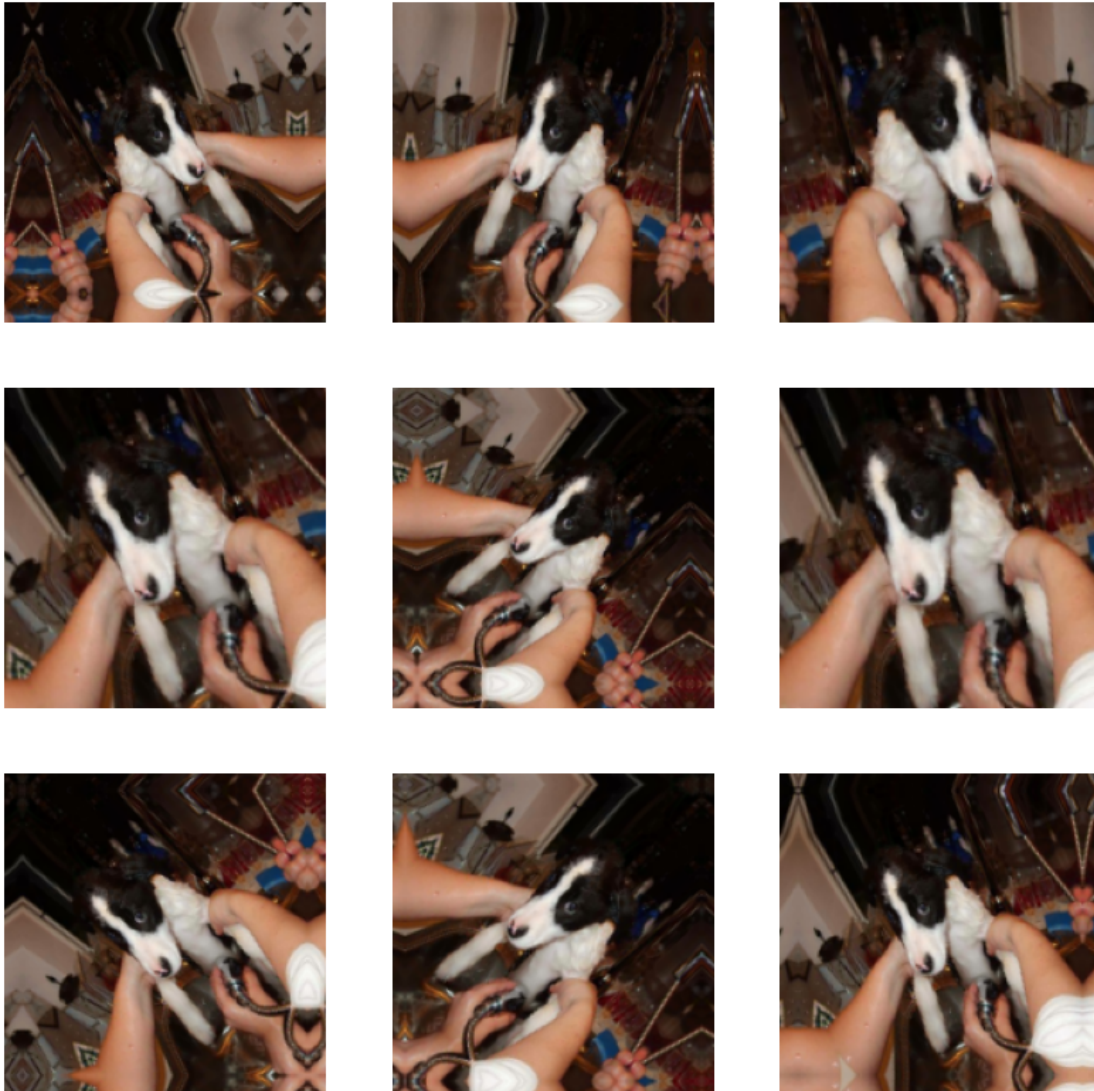
```

```

registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while loop for converting RngReadAndSkip cause there

```

is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.



```
[13]: # model buildup
inputs = keras.Input(shape=(180, 180, 3))
x = data_augmentation(inputs)
x = layers.Rescaling(1./255)(x)
x = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
```

```

x = layers.Flatten()(x)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs=inputs, outputs=outputs)

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

```

```

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3

```

cause there is no registered converter for this op.

```
[14]: # Training the regularized convnet
callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="convnet_from_scratch_with_augmentation_regularization_hw2.
↳keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=150,
    validation_data=validation_dataset,
    callbacks=callbacks)
```

Epoch 1/150

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there

```

is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
250/250 [=====] - 44s 143ms/step - loss: 0.7012 -
accuracy: 0.5346 - val_loss: 0.6612 - val_accuracy: 0.6250
Epoch 2/150
250/250 [=====] - 34s 136ms/step - loss: 0.6899 -
accuracy: 0.6126 - val_loss: 0.6449 - val_accuracy: 0.6310
Epoch 3/150
250/250 [=====] - 34s 137ms/step - loss: 0.6288 -
accuracy: 0.6576 - val_loss: 0.5617 - val_accuracy: 0.7070
Epoch 4/150
250/250 [=====] - 34s 136ms/step - loss: 0.5936 -
accuracy: 0.6916 - val_loss: 0.5214 - val_accuracy: 0.7470
Epoch 5/150
250/250 [=====] - 34s 136ms/step - loss: 0.5664 -
accuracy: 0.7168 - val_loss: 0.5013 - val_accuracy: 0.7470
Epoch 6/150
250/250 [=====] - 34s 134ms/step - loss: 0.5402 -
accuracy: 0.7315 - val_loss: 0.4760 - val_accuracy: 0.7650
Epoch 7/150
250/250 [=====] - 34s 135ms/step - loss: 0.5144 -
accuracy: 0.7540 - val_loss: 0.4437 - val_accuracy: 0.7990
Epoch 8/150
250/250 [=====] - 33s 134ms/step - loss: 0.4954 -
accuracy: 0.7628 - val_loss: 0.3941 - val_accuracy: 0.8140
Epoch 9/150
250/250 [=====] - 34s 136ms/step - loss: 0.4733 -
accuracy: 0.7765 - val_loss: 0.3985 - val_accuracy: 0.8060
Epoch 10/150
250/250 [=====] - 33s 134ms/step - loss: 0.4613 -
accuracy: 0.7839 - val_loss: 0.5060 - val_accuracy: 0.7460
Epoch 11/150
250/250 [=====] - 33s 132ms/step - loss: 0.4443 -
accuracy: 0.7966 - val_loss: 0.3498 - val_accuracy: 0.8390
Epoch 12/150
250/250 [=====] - 33s 133ms/step - loss: 0.4237 -
accuracy: 0.8102 - val_loss: 0.3947 - val_accuracy: 0.8180
Epoch 13/150
250/250 [=====] - 35s 140ms/step - loss: 0.4073 -
accuracy: 0.8198 - val_loss: 0.4941 - val_accuracy: 0.7820
Epoch 14/150

```


250/250 [=====] - 35s 139ms/step - loss: 0.4057 - accuracy: 0.8227 - val_loss: 0.3624 - val_accuracy: 0.8300
Epoch 15/150
250/250 [=====] - 34s 136ms/step - loss: 0.3961 - accuracy: 0.8271 - val_loss: 0.3115 - val_accuracy: 0.8640
Epoch 16/150
250/250 [=====] - 34s 136ms/step - loss: 0.3792 - accuracy: 0.8326 - val_loss: 0.3636 - val_accuracy: 0.8350
Epoch 17/150
250/250 [=====] - 34s 136ms/step - loss: 0.3666 - accuracy: 0.8385 - val_loss: 0.3008 - val_accuracy: 0.8700
Epoch 18/150
250/250 [=====] - 34s 135ms/step - loss: 0.3612 - accuracy: 0.8405 - val_loss: 0.3066 - val_accuracy: 0.8810
Epoch 19/150
250/250 [=====] - 34s 136ms/step - loss: 0.3535 - accuracy: 0.8478 - val_loss: 0.3349 - val_accuracy: 0.8520
Epoch 20/150
250/250 [=====] - 34s 135ms/step - loss: 0.3520 - accuracy: 0.8510 - val_loss: 0.3917 - val_accuracy: 0.8330
Epoch 21/150
250/250 [=====] - 34s 135ms/step - loss: 0.3527 - accuracy: 0.8511 - val_loss: 0.3517 - val_accuracy: 0.8580
Epoch 22/150
250/250 [=====] - 34s 134ms/step - loss: 0.3484 - accuracy: 0.8553 - val_loss: 0.2737 - val_accuracy: 0.8810
Epoch 23/150
250/250 [=====] - 34s 136ms/step - loss: 0.3581 - accuracy: 0.8534 - val_loss: 0.3574 - val_accuracy: 0.8390
Epoch 24/150
250/250 [=====] - 35s 138ms/step - loss: 0.3548 - accuracy: 0.8508 - val_loss: 0.4107 - val_accuracy: 0.8430
Epoch 25/150
250/250 [=====] - 34s 138ms/step - loss: 0.3600 - accuracy: 0.8454 - val_loss: 0.3219 - val_accuracy: 0.8620
Epoch 26/150
250/250 [=====] - 34s 135ms/step - loss: 0.3510 - accuracy: 0.8518 - val_loss: 0.2686 - val_accuracy: 0.8840
Epoch 27/150
250/250 [=====] - 34s 135ms/step - loss: 0.3571 - accuracy: 0.8524 - val_loss: 0.5640 - val_accuracy: 0.7430
Epoch 28/150
250/250 [=====] - 34s 136ms/step - loss: 0.3731 - accuracy: 0.8393 - val_loss: 0.2726 - val_accuracy: 0.8820
Epoch 29/150
250/250 [=====] - 34s 135ms/step - loss: 0.3451 - accuracy: 0.8481 - val_loss: 0.3213 - val_accuracy: 0.8710
Epoch 30/150

250/250 [=====] - 33s 133ms/step - loss: 0.3611 - accuracy: 0.8490 - val_loss: 0.2485 - val_accuracy: 0.8950
Epoch 31/150
250/250 [=====] - 33s 133ms/step - loss: 0.3626 - accuracy: 0.8462 - val_loss: 0.2786 - val_accuracy: 0.8780
Epoch 32/150
250/250 [=====] - 34s 134ms/step - loss: 0.3644 - accuracy: 0.8496 - val_loss: 0.3129 - val_accuracy: 0.8680
Epoch 33/150
250/250 [=====] - 33s 133ms/step - loss: 0.3725 - accuracy: 0.8504 - val_loss: 0.3716 - val_accuracy: 0.8390
Epoch 34/150
250/250 [=====] - 34s 137ms/step - loss: 0.3638 - accuracy: 0.8461 - val_loss: 0.3213 - val_accuracy: 0.8520
Epoch 35/150
250/250 [=====] - 34s 136ms/step - loss: 0.3554 - accuracy: 0.8470 - val_loss: 0.3392 - val_accuracy: 0.8550
Epoch 36/150
250/250 [=====] - 34s 134ms/step - loss: 0.3740 - accuracy: 0.8440 - val_loss: 0.3241 - val_accuracy: 0.8670
Epoch 37/150
250/250 [=====] - 34s 134ms/step - loss: 0.3767 - accuracy: 0.8435 - val_loss: 0.2783 - val_accuracy: 0.8910
Epoch 38/150
250/250 [=====] - 34s 135ms/step - loss: 0.3861 - accuracy: 0.8363 - val_loss: 0.2510 - val_accuracy: 0.8940
Epoch 39/150
250/250 [=====] - 34s 135ms/step - loss: 0.3823 - accuracy: 0.8395 - val_loss: 0.3252 - val_accuracy: 0.8720
Epoch 40/150
250/250 [=====] - 33s 133ms/step - loss: 0.3967 - accuracy: 0.8340 - val_loss: 0.4215 - val_accuracy: 0.8080
Epoch 41/150
250/250 [=====] - 34s 135ms/step - loss: 0.3987 - accuracy: 0.8310 - val_loss: 0.7954 - val_accuracy: 0.8100
Epoch 42/150
250/250 [=====] - 34s 134ms/step - loss: 0.3986 - accuracy: 0.8346 - val_loss: 0.4844 - val_accuracy: 0.7590
Epoch 43/150
250/250 [=====] - 33s 133ms/step - loss: 0.4204 - accuracy: 0.8216 - val_loss: 0.3292 - val_accuracy: 0.8780
Epoch 44/150
250/250 [=====] - 35s 139ms/step - loss: 0.4384 - accuracy: 0.8198 - val_loss: 0.2667 - val_accuracy: 0.8840
Epoch 45/150
250/250 [=====] - 34s 137ms/step - loss: 0.4171 - accuracy: 0.8223 - val_loss: 0.3053 - val_accuracy: 0.8730
Epoch 46/150

250/250 [=====] - 34s 137ms/step - loss: 0.4533 - accuracy: 0.8033 - val_loss: 0.2828 - val_accuracy: 0.8820
Epoch 47/150
250/250 [=====] - 34s 134ms/step - loss: 0.4710 - accuracy: 0.7922 - val_loss: 0.3409 - val_accuracy: 0.8630
Epoch 48/150
250/250 [=====] - 34s 135ms/step - loss: 0.4894 - accuracy: 0.8011 - val_loss: 0.6350 - val_accuracy: 0.8140
Epoch 49/150
250/250 [=====] - 34s 134ms/step - loss: 0.5111 - accuracy: 0.7972 - val_loss: 0.4943 - val_accuracy: 0.7690
Epoch 50/150
250/250 [=====] - 33s 133ms/step - loss: 0.4867 - accuracy: 0.7970 - val_loss: 0.4196 - val_accuracy: 0.8060
Epoch 51/150
250/250 [=====] - 34s 134ms/step - loss: 0.5261 - accuracy: 0.8009 - val_loss: 0.4472 - val_accuracy: 0.7900
Epoch 52/150
250/250 [=====] - 34s 134ms/step - loss: 0.5517 - accuracy: 0.7853 - val_loss: 0.5980 - val_accuracy: 0.7040
Epoch 53/150
250/250 [=====] - 33s 133ms/step - loss: 0.5061 - accuracy: 0.7820 - val_loss: 0.4941 - val_accuracy: 0.8150
Epoch 54/150
250/250 [=====] - 33s 133ms/step - loss: 0.5167 - accuracy: 0.7889 - val_loss: 0.3245 - val_accuracy: 0.8640
Epoch 55/150
250/250 [=====] - 34s 136ms/step - loss: 0.5167 - accuracy: 0.7885 - val_loss: 0.3775 - val_accuracy: 0.8540
Epoch 56/150
250/250 [=====] - 35s 138ms/step - loss: 0.5006 - accuracy: 0.7810 - val_loss: 0.5979 - val_accuracy: 0.7050
Epoch 57/150
250/250 [=====] - 34s 136ms/step - loss: 0.5557 - accuracy: 0.7781 - val_loss: 0.3963 - val_accuracy: 0.8360
Epoch 58/150
250/250 [=====] - 34s 134ms/step - loss: 0.5698 - accuracy: 0.7663 - val_loss: 1.4211 - val_accuracy: 0.5270
Epoch 59/150
250/250 [=====] - 34s 135ms/step - loss: 0.5590 - accuracy: 0.7641 - val_loss: 0.3814 - val_accuracy: 0.8380
Epoch 60/150
250/250 [=====] - 34s 135ms/step - loss: 0.7006 - accuracy: 0.7501 - val_loss: 0.6229 - val_accuracy: 0.6760
Epoch 61/150
250/250 [=====] - 34s 134ms/step - loss: 0.5823 - accuracy: 0.7729 - val_loss: 0.3111 - val_accuracy: 0.8660
Epoch 62/150

250/250 [=====] - 34s 135ms/step - loss: 0.5925 - accuracy: 0.7799 - val_loss: 0.3961 - val_accuracy: 0.8520
Epoch 63/150
250/250 [=====] - 34s 135ms/step - loss: 0.5242 - accuracy: 0.7765 - val_loss: 0.5169 - val_accuracy: 0.7420
Epoch 64/150
250/250 [=====] - 33s 134ms/step - loss: 0.5777 - accuracy: 0.7541 - val_loss: 0.3781 - val_accuracy: 0.8310
Epoch 65/150
250/250 [=====] - 35s 138ms/step - loss: 0.5499 - accuracy: 0.7768 - val_loss: 0.4372 - val_accuracy: 0.7920
Epoch 66/150
250/250 [=====] - 35s 138ms/step - loss: 0.6182 - accuracy: 0.7462 - val_loss: 0.4675 - val_accuracy: 0.7820
Epoch 67/150
250/250 [=====] - 34s 136ms/step - loss: 0.6237 - accuracy: 0.7514 - val_loss: 0.3581 - val_accuracy: 0.8360
Epoch 68/150
250/250 [=====] - 34s 135ms/step - loss: 0.5452 - accuracy: 0.7605 - val_loss: 0.3167 - val_accuracy: 0.8680
Epoch 69/150
250/250 [=====] - 34s 134ms/step - loss: 0.6837 - accuracy: 0.7541 - val_loss: 0.7223 - val_accuracy: 0.6690
Epoch 70/150
250/250 [=====] - 34s 134ms/step - loss: 0.5650 - accuracy: 0.7592 - val_loss: 0.4322 - val_accuracy: 0.8180
Epoch 71/150
250/250 [=====] - 34s 135ms/step - loss: 0.5830 - accuracy: 0.7692 - val_loss: 0.4368 - val_accuracy: 0.7920
Epoch 72/150
250/250 [=====] - 34s 135ms/step - loss: 0.6363 - accuracy: 0.7617 - val_loss: 0.4498 - val_accuracy: 0.8200
Epoch 73/150
250/250 [=====] - 34s 134ms/step - loss: 0.6214 - accuracy: 0.7480 - val_loss: 0.3717 - val_accuracy: 0.8450
Epoch 74/150
250/250 [=====] - 33s 134ms/step - loss: 0.5512 - accuracy: 0.7545 - val_loss: 0.3839 - val_accuracy: 0.8330
Epoch 75/150
250/250 [=====] - 35s 139ms/step - loss: 0.6249 - accuracy: 0.7517 - val_loss: 0.3577 - val_accuracy: 0.8450
Epoch 76/150
250/250 [=====] - 34s 135ms/step - loss: 0.6439 - accuracy: 0.7186 - val_loss: 0.4629 - val_accuracy: 0.7860
Epoch 77/150
250/250 [=====] - 34s 135ms/step - loss: 0.5997 - accuracy: 0.7548 - val_loss: 0.4134 - val_accuracy: 0.8160
Epoch 78/150

250/250 [=====] - 35s 138ms/step - loss: 0.5805 - accuracy: 0.7389 - val_loss: 0.4759 - val_accuracy: 0.7790
Epoch 79/150
250/250 [=====] - 33s 133ms/step - loss: 0.5904 - accuracy: 0.7552 - val_loss: 0.4497 - val_accuracy: 0.8350
Epoch 80/150
250/250 [=====] - 34s 138ms/step - loss: 0.5800 - accuracy: 0.7455 - val_loss: 0.3942 - val_accuracy: 0.8230
Epoch 81/150
250/250 [=====] - 33s 133ms/step - loss: 0.6552 - accuracy: 0.7551 - val_loss: 0.4426 - val_accuracy: 0.7890
Epoch 82/150
250/250 [=====] - 33s 133ms/step - loss: 0.5888 - accuracy: 0.7409 - val_loss: 0.5805 - val_accuracy: 0.6860
Epoch 83/150
250/250 [=====] - 34s 134ms/step - loss: 0.5976 - accuracy: 0.7436 - val_loss: 0.4436 - val_accuracy: 0.8540
Epoch 84/150
250/250 [=====] - 34s 134ms/step - loss: 0.7113 - accuracy: 0.7286 - val_loss: 0.4253 - val_accuracy: 0.8030
Epoch 85/150
250/250 [=====] - 33s 134ms/step - loss: 0.6225 - accuracy: 0.7372 - val_loss: 0.6203 - val_accuracy: 0.7490
Epoch 86/150
250/250 [=====] - 34s 135ms/step - loss: 0.7631 - accuracy: 0.7376 - val_loss: 0.5271 - val_accuracy: 0.7780
Epoch 87/150
250/250 [=====] - 34s 137ms/step - loss: 0.5870 - accuracy: 0.7519 - val_loss: 0.4102 - val_accuracy: 0.8050
Epoch 88/150
250/250 [=====] - 35s 139ms/step - loss: 0.5957 - accuracy: 0.7551 - val_loss: 0.4547 - val_accuracy: 0.7680
Epoch 89/150
250/250 [=====] - 34s 136ms/step - loss: 0.6400 - accuracy: 0.7381 - val_loss: 0.4883 - val_accuracy: 0.7690
Epoch 90/150
250/250 [=====] - 34s 134ms/step - loss: 0.6274 - accuracy: 0.7409 - val_loss: 0.4682 - val_accuracy: 0.7760
Epoch 91/150
250/250 [=====] - 34s 136ms/step - loss: 0.7212 - accuracy: 0.7474 - val_loss: 0.5729 - val_accuracy: 0.7050
Epoch 92/150
250/250 [=====] - 34s 135ms/step - loss: 0.5660 - accuracy: 0.7291 - val_loss: 0.5983 - val_accuracy: 0.7860
Epoch 93/150
250/250 [=====] - 34s 135ms/step - loss: 0.8263 - accuracy: 0.7325 - val_loss: 0.4932 - val_accuracy: 0.7880
Epoch 94/150

250/250 [=====] - 34s 134ms/step - loss: 0.6045 -
accuracy: 0.7255 - val_loss: 0.5526 - val_accuracy: 0.7520
Epoch 95/150
250/250 [=====] - 34s 134ms/step - loss: 0.7676 -
accuracy: 0.7210 - val_loss: 0.4297 - val_accuracy: 0.8020
Epoch 96/150
250/250 [=====] - 33s 134ms/step - loss: 0.6573 -
accuracy: 0.7399 - val_loss: 0.4981 - val_accuracy: 0.7640
Epoch 97/150
250/250 [=====] - 34s 136ms/step - loss: 0.6924 -
accuracy: 0.7359 - val_loss: 0.4568 - val_accuracy: 0.7980
Epoch 98/150
250/250 [=====] - 35s 138ms/step - loss: 0.6243 -
accuracy: 0.7426 - val_loss: 0.4285 - val_accuracy: 0.8110
Epoch 99/150
250/250 [=====] - 34s 137ms/step - loss: 0.6078 -
accuracy: 0.7325 - val_loss: 0.4205 - val_accuracy: 0.7920
Epoch 100/150
250/250 [=====] - 34s 136ms/step - loss: 0.6351 -
accuracy: 0.7258 - val_loss: 0.4391 - val_accuracy: 0.8120
Epoch 101/150
250/250 [=====] - 34s 135ms/step - loss: 0.8014 -
accuracy: 0.7309 - val_loss: 0.3996 - val_accuracy: 0.8230
Epoch 102/150
250/250 [=====] - 34s 136ms/step - loss: 0.6331 -
accuracy: 0.7057 - val_loss: 0.5286 - val_accuracy: 0.7430
Epoch 103/150
250/250 [=====] - 34s 134ms/step - loss: 0.8561 -
accuracy: 0.7089 - val_loss: 0.5470 - val_accuracy: 0.7330
Epoch 104/150
250/250 [=====] - 34s 135ms/step - loss: 1.2972 -
accuracy: 0.7100 - val_loss: 0.4445 - val_accuracy: 0.8160
Epoch 105/150
250/250 [=====] - 34s 135ms/step - loss: 0.8530 -
accuracy: 0.7014 - val_loss: 0.5205 - val_accuracy: 0.7710
Epoch 106/150
250/250 [=====] - 33s 131ms/step - loss: 0.7128 -
accuracy: 0.7171 - val_loss: 0.5601 - val_accuracy: 0.7260
Epoch 107/150
250/250 [=====] - 34s 136ms/step - loss: 0.6173 -
accuracy: 0.7160 - val_loss: 0.4319 - val_accuracy: 0.8080
Epoch 108/150
250/250 [=====] - 34s 138ms/step - loss: 0.6807 -
accuracy: 0.7026 - val_loss: 0.5683 - val_accuracy: 0.7210
Epoch 109/150
250/250 [=====] - 35s 139ms/step - loss: 0.7384 -
accuracy: 0.6994 - val_loss: 0.4678 - val_accuracy: 0.7860
Epoch 110/150

250/250 [=====] - 34s 137ms/step - loss: 0.6008 - accuracy: 0.7069 - val_loss: 0.4824 - val_accuracy: 0.7550
Epoch 111/150
250/250 [=====] - 34s 135ms/step - loss: 0.6991 - accuracy: 0.7066 - val_loss: 0.5010 - val_accuracy: 0.7730
Epoch 112/150
250/250 [=====] - 34s 136ms/step - loss: 0.6793 - accuracy: 0.7004 - val_loss: 0.4272 - val_accuracy: 0.8180
Epoch 113/150
250/250 [=====] - 34s 134ms/step - loss: 0.6241 - accuracy: 0.7117 - val_loss: 0.7522 - val_accuracy: 0.8020
Epoch 114/150
250/250 [=====] - 33s 134ms/step - loss: 0.6940 - accuracy: 0.7215 - val_loss: 0.4977 - val_accuracy: 0.7820
Epoch 115/150
250/250 [=====] - 34s 135ms/step - loss: 0.6115 - accuracy: 0.7147 - val_loss: 0.5043 - val_accuracy: 0.7610
Epoch 116/150
250/250 [=====] - 34s 135ms/step - loss: 0.6538 - accuracy: 0.6892 - val_loss: 0.5213 - val_accuracy: 0.7610
Epoch 117/150
250/250 [=====] - 33s 133ms/step - loss: 0.7095 - accuracy: 0.6840 - val_loss: 0.6021 - val_accuracy: 0.6760
Epoch 118/150
250/250 [=====] - 34s 135ms/step - loss: 0.6744 - accuracy: 0.7025 - val_loss: 0.6466 - val_accuracy: 0.7920
Epoch 119/150
250/250 [=====] - 34s 136ms/step - loss: 0.7859 - accuracy: 0.6984 - val_loss: 0.7839 - val_accuracy: 0.7810
Epoch 120/150
250/250 [=====] - 35s 139ms/step - loss: 0.6979 - accuracy: 0.6935 - val_loss: 0.6910 - val_accuracy: 0.7240
Epoch 121/150
250/250 [=====] - 34s 137ms/step - loss: 0.7309 - accuracy: 0.6837 - val_loss: 0.5459 - val_accuracy: 0.7600
Epoch 122/150
250/250 [=====] - 34s 135ms/step - loss: 0.6762 - accuracy: 0.7016 - val_loss: 0.5654 - val_accuracy: 0.7780
Epoch 123/150
250/250 [=====] - 34s 135ms/step - loss: 0.7470 - accuracy: 0.6821 - val_loss: 0.6197 - val_accuracy: 0.5820
Epoch 124/150
250/250 [=====] - 34s 134ms/step - loss: 0.7507 - accuracy: 0.6781 - val_loss: 0.4930 - val_accuracy: 0.7990
Epoch 125/150
250/250 [=====] - 34s 134ms/step - loss: 0.6865 - accuracy: 0.7168 - val_loss: 0.4814 - val_accuracy: 0.7650
Epoch 126/150

250/250 [=====] - 33s 133ms/step - loss: 0.6502 - accuracy: 0.7034 - val_loss: 0.4936 - val_accuracy: 0.7950
Epoch 127/150
250/250 [=====] - 33s 132ms/step - loss: 0.7507 - accuracy: 0.7196 - val_loss: 0.5833 - val_accuracy: 0.6800
Epoch 128/150
250/250 [=====] - 33s 132ms/step - loss: 0.6374 - accuracy: 0.6815 - val_loss: 0.4913 - val_accuracy: 0.7790
Epoch 129/150
250/250 [=====] - 34s 136ms/step - loss: 0.6974 - accuracy: 0.6873 - val_loss: 0.5636 - val_accuracy: 0.6910
Epoch 130/150
250/250 [=====] - 34s 135ms/step - loss: 0.6870 - accuracy: 0.6693 - val_loss: 0.4807 - val_accuracy: 0.7740
Epoch 131/150
250/250 [=====] - 34s 135ms/step - loss: 0.7427 - accuracy: 0.7035 - val_loss: 0.5986 - val_accuracy: 0.6720
Epoch 132/150
250/250 [=====] - 34s 135ms/step - loss: 0.9072 - accuracy: 0.6693 - val_loss: 0.5780 - val_accuracy: 0.6920
Epoch 133/150
250/250 [=====] - 34s 135ms/step - loss: 1.0831 - accuracy: 0.7029 - val_loss: 4.4167 - val_accuracy: 0.5730
Epoch 134/150
250/250 [=====] - 34s 135ms/step - loss: 0.6270 - accuracy: 0.7121 - val_loss: 0.7365 - val_accuracy: 0.7130
Epoch 135/150
250/250 [=====] - 33s 133ms/step - loss: 0.7493 - accuracy: 0.6941 - val_loss: 0.4669 - val_accuracy: 0.7650
Epoch 136/150
250/250 [=====] - 33s 134ms/step - loss: 0.5666 - accuracy: 0.7219 - val_loss: 0.5057 - val_accuracy: 0.7540
Epoch 137/150
250/250 [=====] - 34s 134ms/step - loss: 0.6211 - accuracy: 0.7025 - val_loss: 0.6238 - val_accuracy: 0.7650
Epoch 138/150
250/250 [=====] - 34s 134ms/step - loss: 0.7702 - accuracy: 0.6790 - val_loss: 0.5549 - val_accuracy: 0.7300
Epoch 139/150
250/250 [=====] - 33s 133ms/step - loss: 0.6590 - accuracy: 0.6977 - val_loss: 0.6518 - val_accuracy: 0.7080
Epoch 140/150
250/250 [=====] - 34s 134ms/step - loss: 0.7350 - accuracy: 0.6905 - val_loss: 0.5814 - val_accuracy: 0.7230
Epoch 141/150
250/250 [=====] - 34s 136ms/step - loss: 0.7303 - accuracy: 0.7071 - val_loss: 1.7395 - val_accuracy: 0.7780
Epoch 142/150


```

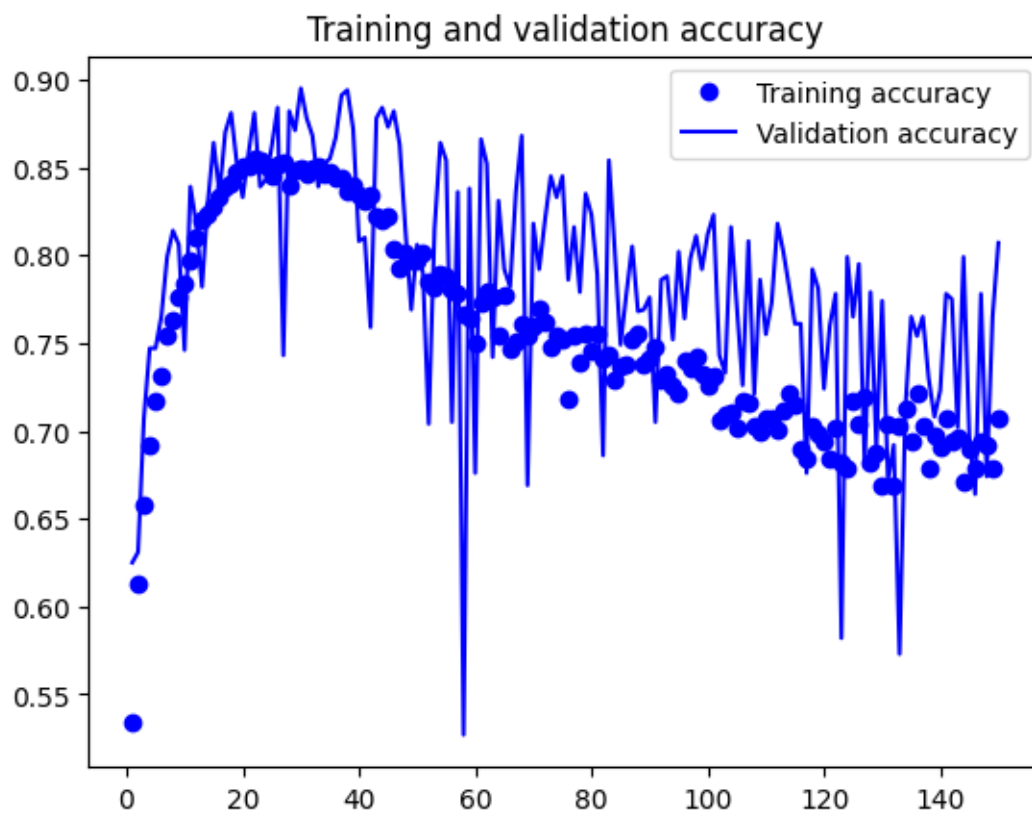
250/250 [=====] - 34s 136ms/step - loss: 0.7347 -
accuracy: 0.6945 - val_loss: 0.4561 - val_accuracy: 0.7750
Epoch 143/150
250/250 [=====] - 34s 135ms/step - loss: 0.7408 -
accuracy: 0.6960 - val_loss: 0.5582 - val_accuracy: 0.7020
Epoch 144/150
250/250 [=====] - 34s 135ms/step - loss: 0.6952 -
accuracy: 0.6714 - val_loss: 0.5449 - val_accuracy: 0.7990
Epoch 145/150
250/250 [=====] - 34s 134ms/step - loss: 0.8853 -
accuracy: 0.6898 - val_loss: 0.5713 - val_accuracy: 0.6970
Epoch 146/150
250/250 [=====] - 33s 134ms/step - loss: 0.7648 -
accuracy: 0.6791 - val_loss: 0.5919 - val_accuracy: 0.6640
Epoch 147/150
250/250 [=====] - 33s 131ms/step - loss: 0.6385 -
accuracy: 0.6939 - val_loss: 0.6526 - val_accuracy: 0.7780
Epoch 148/150
250/250 [=====] - 33s 132ms/step - loss: 0.6832 -
accuracy: 0.6914 - val_loss: 0.7472 - val_accuracy: 0.6740
Epoch 149/150
250/250 [=====] - 33s 132ms/step - loss: 0.7747 -
accuracy: 0.6783 - val_loss: 0.4979 - val_accuracy: 0.7650
Epoch 150/150
250/250 [=====] - 34s 134ms/step - loss: 0.7180 -
accuracy: 0.7074 - val_loss: 0.6532 - val_accuracy: 0.8070

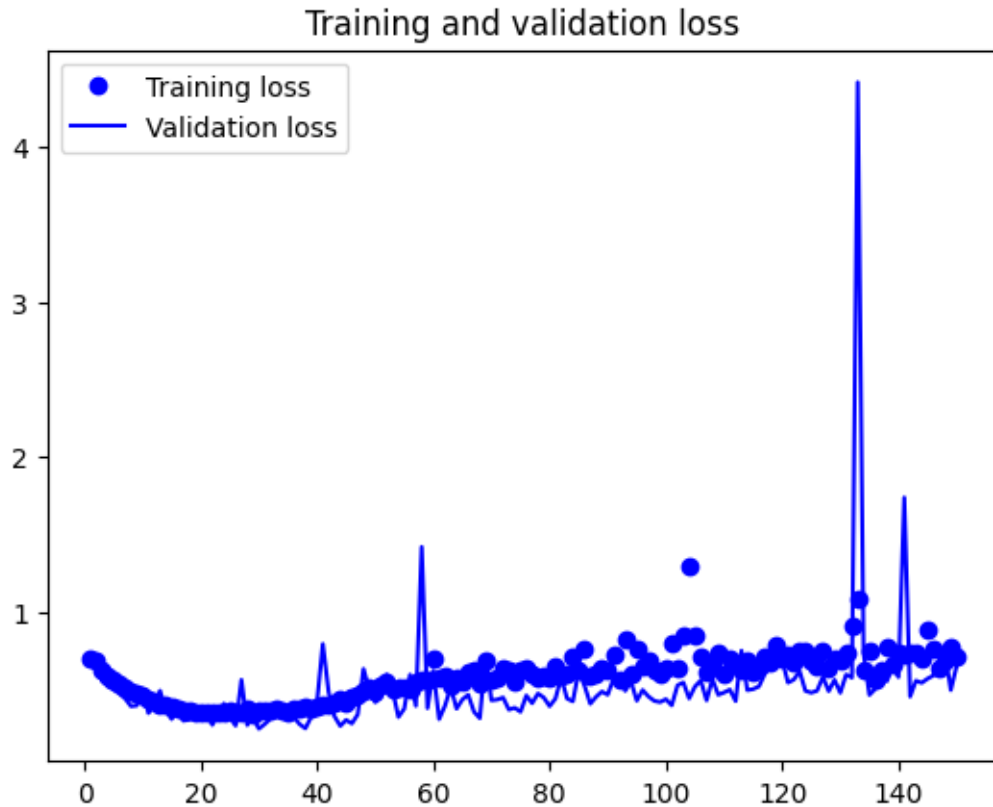
```

```

[15]: # plot the performance
import matplotlib.pyplot as plt
accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
loss = history.history["loss"]
val_loss = history.history["val_loss"]
epochs = range(1, len(accuracy) + 1)
plt.plot(epochs, accuracy, "bo", label="Training accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.legend()
plt.figure()
plt.plot(epochs, loss, "bo", label="Training loss")
plt.plot(epochs, val_loss, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.legend()
plt.show()

```





```
[16]: # Evaluating the model on the test set
```

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

```
32/32 [=====] - 1s 16ms/step - loss: 0.4997 - accuracy: 0.7890
```

```
Test accuracy: 0.789
```

```
[ ]: # Based on the previous part1 and 2, I am assuming that the best epoch should
      ↳ be within 50, so I will not
      # do the epoch to 150 but reduce it to 100 for saving time

      # PART3, FINAL TUNING ON SCRATCH MODEL
```

```
[17]: original_dir = pathlib.Path('C:\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\train')
      new_base_dir = pathlib.Path('C:
      ↳ \\Users\\zhong\\Desktop\\FA23_DL\\HW2\\cats_vs_dogs_hw2_p3_tuning_scratch')
```

```

def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = new_base_dir / subset_name / category
        os.makedirs(dir)
        fnames = [f"{category}.{i}.jpg"
                   for i in range(start_index, end_index)]
        for fname in fnames:
            shutil.copyfile(src=original_dir / fname,
                            dst=dir / fname)
make_subset("train", start_index=0, end_index=1500)          ###1500  ␣
↪###      1500 3000
make_subset("validation", start_index=1500, end_index=2000) ###500
make_subset("test", start_index=2000, end_index=2500)       ###500 TEST

```

```

[18]: from tensorflow import keras
      from tensorflow.keras import layers

```

```

[19]: from tensorflow.keras.utils import image_dataset_from_directory

train_dataset = image_dataset_from_directory(
    new_base_dir / "train",
    image_size=(180, 180),
    batch_size=32)
validation_dataset = image_dataset_from_directory(
    new_base_dir / "validation",
    image_size=(180, 180),
    batch_size=32)
test_dataset = image_dataset_from_directory(
    new_base_dir / "test",
    image_size=(180, 180),
    batch_size=32)

```

Found 3000 files belonging to 2 classes.

Found 1000 files belonging to 2 classes.

Found 1000 files belonging to 2 classes.

```

[20]: import numpy as np
      import tensorflow as tf
      random_numbers = np.random.normal(size=(1000, 16))
      dataset = tf.data.Dataset.from_tensor_slices(random_numbers)

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

```

```

batched_dataset = dataset.batch(32)
for i, element in enumerate(batched_dataset):
    print(element.shape)
    if i >= 2:
        break
reshaped_dataset = dataset.map(lambda x: tf.reshape(x, (4, 4)))
for i, element in enumerate(reshaped_dataset):
    print(element.shape)
    if i >= 2:
        break
#Displaying the shapes of the data and labels yielded by the Dataset
for data_batch, labels_batch in train_dataset:
    print("data batch shape:", data_batch.shape)
    print("labels batch shape:", labels_batch.shape)
    break

```

```

(16,)
(16,)
(16,)
(32, 16)
(32, 16)
(32, 16)
(4, 4)
(4, 4)
(4, 4)
data batch shape: (32, 180, 180, 3)
labels batch shape: (32,)

```

```

[21]: data_augmentation = keras.Sequential(
    [
        layers.RandomFlip("horizontal"),
        layers.RandomRotation(0.15),
        layers.RandomZoom(0.25),
    ]
)

```

```

[22]: plt.figure(figsize=(10, 10))
for images, _ in train_dataset.take(1):
    for i in range(9):
        augmented_images = data_augmentation(images)
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(augmented_images[0].numpy().astype("uint8"))
        plt.axis("off")

```



```
[23]: # model buildup
inputs = keras.Input(shape=(180, 180, 3))
x = data_augmentation(inputs)
x = layers.Rescaling(1./255)(x)
x = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
```

```

x = layers.Flatten()(x)
x = layers.Dropout(0.4)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs=inputs, outputs=outputs)

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

```

```

[25]: # Training the regularized convnet
callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="convnet_from_scratch_with_augmentation_regularization_hw2.
↳keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=100,
    validation_data=validation_dataset,
    callbacks=callbacks)

```

Epoch 1/100

94/94 [=====] - 14s 146ms/step - loss: 0.6933 - accuracy: 0.5477 - val_loss: 0.6741 - val_accuracy: 0.5610

Epoch 2/100

94/94 [=====] - 13s 134ms/step - loss: 0.6774 - accuracy: 0.5887 - val_loss: 0.6497 - val_accuracy: 0.6180

Epoch 3/100

94/94 [=====] - 13s 135ms/step - loss: 0.6569 - accuracy: 0.6177 - val_loss: 0.6462 - val_accuracy: 0.6050

Epoch 4/100

94/94 [=====] - 12s 132ms/step - loss: 0.6456 - accuracy: 0.6383 - val_loss: 0.6686 - val_accuracy: 0.6040

Epoch 5/100

94/94 [=====] - 13s 136ms/step - loss: 0.6293 - accuracy: 0.6553 - val_loss: 0.6017 - val_accuracy: 0.6920

Epoch 6/100

94/94 [=====] - 13s 137ms/step - loss: 0.5948 - accuracy: 0.6803 - val_loss: 0.6150 - val_accuracy: 0.6900

Epoch 7/100

94/94 [=====] - 13s 135ms/step - loss: 0.5881 - accuracy: 0.6997 - val_loss: 0.6176 - val_accuracy: 0.6950

Epoch 8/100

94/94 [=====] - 13s 135ms/step - loss: 0.5687 - accuracy: 0.7053 - val_loss: 0.5722 - val_accuracy: 0.7080

Epoch 9/100

94/94 [=====] - 13s 136ms/step - loss: 0.5557 - accuracy: 0.7153 - val_loss: 0.5310 - val_accuracy: 0.7400
Epoch 10/100
94/94 [=====] - 13s 136ms/step - loss: 0.5462 - accuracy: 0.7277 - val_loss: 0.6058 - val_accuracy: 0.7090
Epoch 11/100
94/94 [=====] - 13s 140ms/step - loss: 0.5314 - accuracy: 0.7343 - val_loss: 0.5532 - val_accuracy: 0.7270
Epoch 12/100
94/94 [=====] - 13s 140ms/step - loss: 0.5143 - accuracy: 0.7507 - val_loss: 0.5636 - val_accuracy: 0.7470
Epoch 13/100
94/94 [=====] - 13s 139ms/step - loss: 0.4897 - accuracy: 0.7600 - val_loss: 0.5539 - val_accuracy: 0.7480
Epoch 14/100
94/94 [=====] - 13s 140ms/step - loss: 0.4948 - accuracy: 0.7537 - val_loss: 0.5801 - val_accuracy: 0.7170
Epoch 15/100
94/94 [=====] - 13s 140ms/step - loss: 0.4876 - accuracy: 0.7593 - val_loss: 0.7975 - val_accuracy: 0.6830
Epoch 16/100
94/94 [=====] - 13s 139ms/step - loss: 0.4909 - accuracy: 0.7750 - val_loss: 0.5439 - val_accuracy: 0.7570
Epoch 17/100
94/94 [=====] - 13s 138ms/step - loss: 0.4508 - accuracy: 0.7893 - val_loss: 0.5600 - val_accuracy: 0.7580
Epoch 18/100
94/94 [=====] - 13s 137ms/step - loss: 0.4711 - accuracy: 0.7810 - val_loss: 0.5017 - val_accuracy: 0.7690
Epoch 19/100
94/94 [=====] - 13s 135ms/step - loss: 0.4604 - accuracy: 0.7873 - val_loss: 0.4952 - val_accuracy: 0.7690
Epoch 20/100
94/94 [=====] - 13s 135ms/step - loss: 0.4439 - accuracy: 0.7907 - val_loss: 0.5028 - val_accuracy: 0.7730
Epoch 21/100
94/94 [=====] - 13s 137ms/step - loss: 0.4381 - accuracy: 0.7963 - val_loss: 0.5385 - val_accuracy: 0.7360
Epoch 22/100
94/94 [=====] - 13s 138ms/step - loss: 0.4358 - accuracy: 0.8000 - val_loss: 0.4964 - val_accuracy: 0.7530
Epoch 23/100
94/94 [=====] - 13s 137ms/step - loss: 0.4268 - accuracy: 0.8010 - val_loss: 0.4762 - val_accuracy: 0.8000
Epoch 24/100
94/94 [=====] - 13s 137ms/step - loss: 0.4070 - accuracy: 0.8150 - val_loss: 0.4659 - val_accuracy: 0.7900
Epoch 25/100

94/94 [=====] - 13s 135ms/step - loss: 0.4197 -
accuracy: 0.8067 - val_loss: 0.5255 - val_accuracy: 0.7790
Epoch 26/100
94/94 [=====] - 13s 138ms/step - loss: 0.4185 -
accuracy: 0.8050 - val_loss: 0.4561 - val_accuracy: 0.7990
Epoch 27/100
94/94 [=====] - 13s 137ms/step - loss: 0.3934 -
accuracy: 0.8207 - val_loss: 0.4852 - val_accuracy: 0.7930
Epoch 28/100
94/94 [=====] - 13s 137ms/step - loss: 0.3871 -
accuracy: 0.8347 - val_loss: 0.5005 - val_accuracy: 0.8100
Epoch 29/100
94/94 [=====] - 13s 138ms/step - loss: 0.3872 -
accuracy: 0.8247 - val_loss: 0.6151 - val_accuracy: 0.7710
Epoch 30/100
94/94 [=====] - 13s 138ms/step - loss: 0.3754 -
accuracy: 0.8293 - val_loss: 0.4398 - val_accuracy: 0.8080
Epoch 31/100
94/94 [=====] - 13s 139ms/step - loss: 0.3648 -
accuracy: 0.8473 - val_loss: 0.4202 - val_accuracy: 0.8200
Epoch 32/100
94/94 [=====] - 13s 136ms/step - loss: 0.3567 -
accuracy: 0.8417 - val_loss: 0.4830 - val_accuracy: 0.7820
Epoch 33/100
94/94 [=====] - 13s 134ms/step - loss: 0.3501 -
accuracy: 0.8470 - val_loss: 0.4998 - val_accuracy: 0.8230
Epoch 34/100
94/94 [=====] - 13s 135ms/step - loss: 0.3492 -
accuracy: 0.8460 - val_loss: 0.4432 - val_accuracy: 0.8350
Epoch 35/100
94/94 [=====] - 13s 135ms/step - loss: 0.3484 -
accuracy: 0.8497 - val_loss: 0.4831 - val_accuracy: 0.8060
Epoch 36/100
94/94 [=====] - 13s 134ms/step - loss: 0.3156 -
accuracy: 0.8707 - val_loss: 0.4831 - val_accuracy: 0.8240
Epoch 37/100
94/94 [=====] - 13s 134ms/step - loss: 0.3296 -
accuracy: 0.8613 - val_loss: 0.4160 - val_accuracy: 0.8320
Epoch 38/100
94/94 [=====] - 13s 134ms/step - loss: 0.3342 -
accuracy: 0.8650 - val_loss: 0.4310 - val_accuracy: 0.8380
Epoch 39/100
94/94 [=====] - 13s 135ms/step - loss: 0.3268 -
accuracy: 0.8627 - val_loss: 0.5273 - val_accuracy: 0.8060
Epoch 40/100
94/94 [=====] - 13s 136ms/step - loss: 0.3025 -
accuracy: 0.8733 - val_loss: 0.4937 - val_accuracy: 0.8100
Epoch 41/100

94/94 [=====] - 13s 135ms/step - loss: 0.3021 - accuracy: 0.8657 - val_loss: 0.7591 - val_accuracy: 0.7340
Epoch 42/100
94/94 [=====] - 13s 134ms/step - loss: 0.2910 - accuracy: 0.8797 - val_loss: 0.4616 - val_accuracy: 0.8260
Epoch 43/100
94/94 [=====] - 13s 133ms/step - loss: 0.2996 - accuracy: 0.8787 - val_loss: 0.6178 - val_accuracy: 0.7720
Epoch 44/100
94/94 [=====] - 13s 135ms/step - loss: 0.2847 - accuracy: 0.8777 - val_loss: 0.6774 - val_accuracy: 0.7890
Epoch 45/100
94/94 [=====] - 13s 134ms/step - loss: 0.2884 - accuracy: 0.8787 - val_loss: 0.4773 - val_accuracy: 0.8390
Epoch 46/100
94/94 [=====] - 13s 134ms/step - loss: 0.2751 - accuracy: 0.8830 - val_loss: 0.5585 - val_accuracy: 0.8290
Epoch 47/100
94/94 [=====] - 13s 133ms/step - loss: 0.2799 - accuracy: 0.8853 - val_loss: 0.6199 - val_accuracy: 0.7940
Epoch 48/100
94/94 [=====] - 13s 135ms/step - loss: 0.2588 - accuracy: 0.8967 - val_loss: 0.4893 - val_accuracy: 0.8450
Epoch 49/100
94/94 [=====] - 13s 134ms/step - loss: 0.2759 - accuracy: 0.8773 - val_loss: 0.5140 - val_accuracy: 0.8270
Epoch 50/100
94/94 [=====] - 12s 132ms/step - loss: 0.2825 - accuracy: 0.8923 - val_loss: 0.6363 - val_accuracy: 0.7960
Epoch 51/100
94/94 [=====] - 13s 134ms/step - loss: 0.2778 - accuracy: 0.8833 - val_loss: 0.3966 - val_accuracy: 0.8420
Epoch 52/100
94/94 [=====] - 13s 134ms/step - loss: 0.2578 - accuracy: 0.8960 - val_loss: 0.4066 - val_accuracy: 0.8580
Epoch 53/100
94/94 [=====] - 13s 134ms/step - loss: 0.2607 - accuracy: 0.8967 - val_loss: 0.4410 - val_accuracy: 0.8500
Epoch 54/100
94/94 [=====] - 13s 134ms/step - loss: 0.2815 - accuracy: 0.8990 - val_loss: 0.4674 - val_accuracy: 0.8340
Epoch 55/100
94/94 [=====] - 13s 133ms/step - loss: 0.2623 - accuracy: 0.8967 - val_loss: 0.6562 - val_accuracy: 0.7810
Epoch 56/100
94/94 [=====] - 13s 133ms/step - loss: 0.2434 - accuracy: 0.9003 - val_loss: 0.4583 - val_accuracy: 0.8510
Epoch 57/100

94/94 [=====] - 13s 133ms/step - loss: 0.2697 - accuracy: 0.8947 - val_loss: 0.4220 - val_accuracy: 0.8460
Epoch 58/100
94/94 [=====] - 13s 137ms/step - loss: 0.2476 - accuracy: 0.8973 - val_loss: 0.4528 - val_accuracy: 0.8440
Epoch 59/100
94/94 [=====] - 13s 135ms/step - loss: 0.2534 - accuracy: 0.9077 - val_loss: 0.6215 - val_accuracy: 0.7970
Epoch 60/100
94/94 [=====] - 13s 141ms/step - loss: 0.2501 - accuracy: 0.8950 - val_loss: 0.5519 - val_accuracy: 0.8330
Epoch 61/100
94/94 [=====] - 13s 141ms/step - loss: 0.2512 - accuracy: 0.9010 - val_loss: 0.5365 - val_accuracy: 0.8130
Epoch 62/100
94/94 [=====] - 13s 142ms/step - loss: 0.2554 - accuracy: 0.9037 - val_loss: 0.5130 - val_accuracy: 0.8430
Epoch 63/100
94/94 [=====] - 13s 141ms/step - loss: 0.2366 - accuracy: 0.9067 - val_loss: 0.7231 - val_accuracy: 0.8220
Epoch 64/100
94/94 [=====] - 14s 144ms/step - loss: 0.2375 - accuracy: 0.9103 - val_loss: 0.5547 - val_accuracy: 0.8260
Epoch 65/100
94/94 [=====] - 13s 136ms/step - loss: 0.2404 - accuracy: 0.9070 - val_loss: 0.4721 - val_accuracy: 0.8330
Epoch 66/100
94/94 [=====] - 13s 140ms/step - loss: 0.2391 - accuracy: 0.9070 - val_loss: 0.9232 - val_accuracy: 0.7790
Epoch 67/100
94/94 [=====] - 13s 138ms/step - loss: 0.2582 - accuracy: 0.8990 - val_loss: 0.5851 - val_accuracy: 0.8280
Epoch 68/100
94/94 [=====] - 13s 141ms/step - loss: 0.2188 - accuracy: 0.9150 - val_loss: 1.0971 - val_accuracy: 0.7380
Epoch 69/100
94/94 [=====] - 13s 142ms/step - loss: 0.2425 - accuracy: 0.9020 - val_loss: 0.5287 - val_accuracy: 0.8610
Epoch 70/100
94/94 [=====] - 13s 137ms/step - loss: 0.2301 - accuracy: 0.9157 - val_loss: 0.5102 - val_accuracy: 0.8460
Epoch 71/100
94/94 [=====] - 13s 139ms/step - loss: 0.2097 - accuracy: 0.9243 - val_loss: 0.6185 - val_accuracy: 0.8320
Epoch 72/100
94/94 [=====] - 13s 139ms/step - loss: 0.2403 - accuracy: 0.9157 - val_loss: 0.8641 - val_accuracy: 0.8230
Epoch 73/100

94/94 [=====] - 13s 138ms/step - loss: 0.2496 -
accuracy: 0.9017 - val_loss: 0.6203 - val_accuracy: 0.8460
Epoch 74/100
94/94 [=====] - 13s 138ms/step - loss: 0.2452 -
accuracy: 0.9077 - val_loss: 0.4801 - val_accuracy: 0.8530
Epoch 75/100
94/94 [=====] - 13s 138ms/step - loss: 0.2275 -
accuracy: 0.9140 - val_loss: 0.5473 - val_accuracy: 0.8660
Epoch 76/100
94/94 [=====] - 13s 138ms/step - loss: 0.2394 -
accuracy: 0.9127 - val_loss: 0.7046 - val_accuracy: 0.8330
Epoch 77/100
94/94 [=====] - 13s 137ms/step - loss: 0.2402 -
accuracy: 0.9140 - val_loss: 0.5937 - val_accuracy: 0.8340
Epoch 78/100
94/94 [=====] - 13s 134ms/step - loss: 0.2333 -
accuracy: 0.9067 - val_loss: 0.7694 - val_accuracy: 0.8210
Epoch 79/100
94/94 [=====] - 12s 131ms/step - loss: 0.2357 -
accuracy: 0.9093 - val_loss: 0.6945 - val_accuracy: 0.8140
Epoch 80/100
94/94 [=====] - 13s 141ms/step - loss: 0.1957 -
accuracy: 0.9277 - val_loss: 0.9213 - val_accuracy: 0.8050
Epoch 81/100
94/94 [=====] - 13s 140ms/step - loss: 0.2411 -
accuracy: 0.9147 - val_loss: 0.5927 - val_accuracy: 0.8550
Epoch 82/100
94/94 [=====] - 13s 136ms/step - loss: 0.2067 -
accuracy: 0.9240 - val_loss: 0.5359 - val_accuracy: 0.8420
Epoch 83/100
94/94 [=====] - 13s 140ms/step - loss: 0.2578 -
accuracy: 0.9090 - val_loss: 0.6042 - val_accuracy: 0.8430
Epoch 84/100
94/94 [=====] - 13s 136ms/step - loss: 0.2423 -
accuracy: 0.9147 - val_loss: 0.5030 - val_accuracy: 0.8560
Epoch 85/100
94/94 [=====] - 13s 138ms/step - loss: 0.2317 -
accuracy: 0.9160 - val_loss: 0.8157 - val_accuracy: 0.8220
Epoch 86/100
94/94 [=====] - 13s 135ms/step - loss: 0.2364 -
accuracy: 0.9150 - val_loss: 0.6502 - val_accuracy: 0.8500
Epoch 87/100
94/94 [=====] - 13s 135ms/step - loss: 0.2280 -
accuracy: 0.9250 - val_loss: 0.5147 - val_accuracy: 0.8380
Epoch 88/100
94/94 [=====] - 12s 128ms/step - loss: 0.2250 -
accuracy: 0.9187 - val_loss: 0.8434 - val_accuracy: 0.8230
Epoch 89/100

```

94/94 [=====] - 13s 133ms/step - loss: 0.2049 -
accuracy: 0.9260 - val_loss: 0.7450 - val_accuracy: 0.8360
Epoch 90/100
94/94 [=====] - 13s 136ms/step - loss: 0.2274 -
accuracy: 0.9167 - val_loss: 0.6192 - val_accuracy: 0.8320
Epoch 91/100
94/94 [=====] - 13s 134ms/step - loss: 0.2597 -
accuracy: 0.9080 - val_loss: 0.5392 - val_accuracy: 0.8500
Epoch 92/100
94/94 [=====] - 13s 136ms/step - loss: 0.2077 -
accuracy: 0.9240 - val_loss: 0.8391 - val_accuracy: 0.8450
Epoch 93/100
94/94 [=====] - 13s 140ms/step - loss: 0.2332 -
accuracy: 0.9130 - val_loss: 0.7394 - val_accuracy: 0.8580
Epoch 94/100
94/94 [=====] - 13s 140ms/step - loss: 0.2313 -
accuracy: 0.9130 - val_loss: 0.5032 - val_accuracy: 0.8410
Epoch 95/100
94/94 [=====] - 13s 138ms/step - loss: 0.2345 -
accuracy: 0.9103 - val_loss: 0.6392 - val_accuracy: 0.8580
Epoch 96/100
94/94 [=====] - 13s 138ms/step - loss: 0.2138 -
accuracy: 0.9237 - val_loss: 0.6212 - val_accuracy: 0.8570
Epoch 97/100
94/94 [=====] - 13s 138ms/step - loss: 0.2616 -
accuracy: 0.9100 - val_loss: 0.6348 - val_accuracy: 0.8550
Epoch 98/100
94/94 [=====] - 13s 140ms/step - loss: 0.2224 -
accuracy: 0.9177 - val_loss: 0.8389 - val_accuracy: 0.8590
Epoch 99/100
94/94 [=====] - 13s 139ms/step - loss: 0.2257 -
accuracy: 0.9120 - val_loss: 0.7622 - val_accuracy: 0.8680
Epoch 100/100
94/94 [=====] - 13s 137ms/step - loss: 0.2470 -
accuracy: 0.9177 - val_loss: 0.7912 - val_accuracy: 0.8410

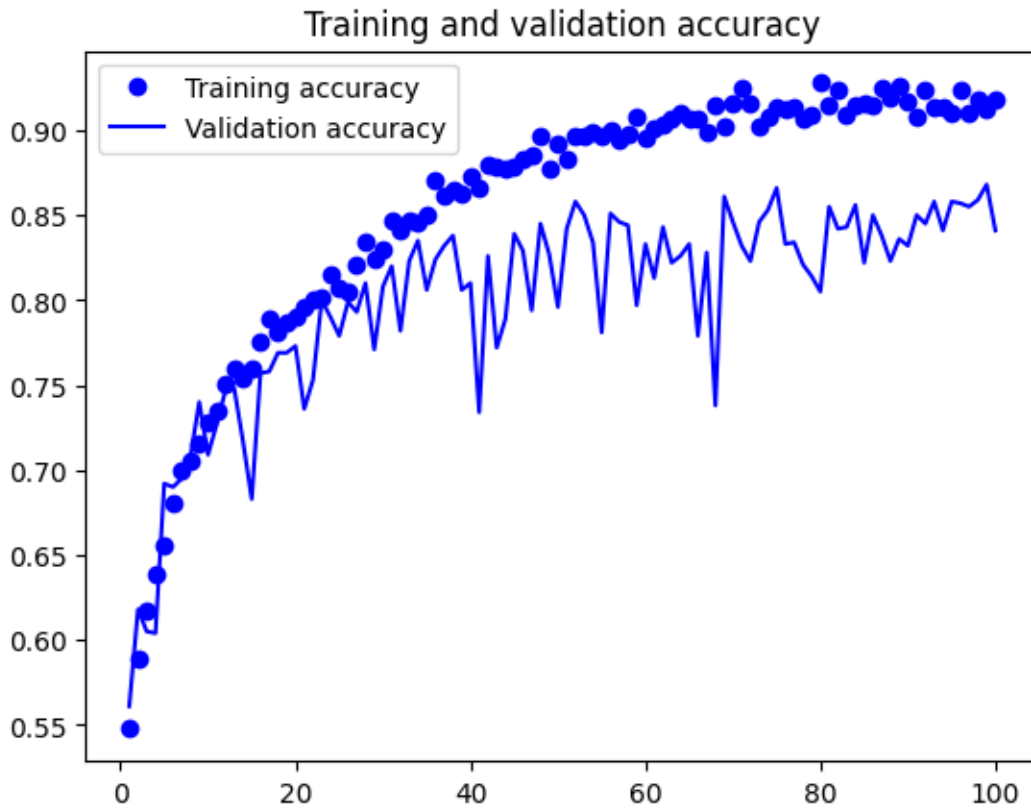
```

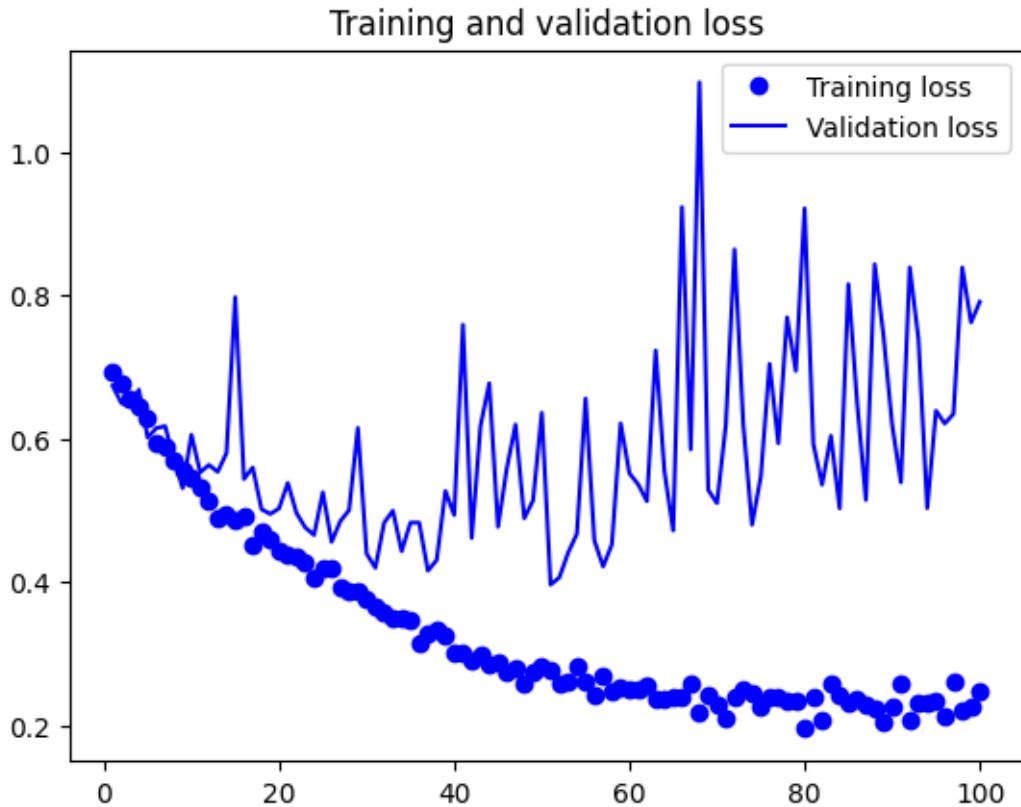
```

[26]: # plot the performance
import matplotlib.pyplot as plt
accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
loss = history.history["loss"]
val_loss = history.history["val_loss"]
epochs = range(1, len(accuracy) + 1)
plt.plot(epochs, accuracy, "bo", label="Training accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.legend()

```

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





```
[27]: # Evaluating the model on the test set
```

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

```
32/32 [=====] - 1s 17ms/step - loss: 0.4813 - accuracy: 0.7860
```

```
Test accuracy: 0.786
```

```
[ ]: # PART3 FINAL TUNING
```

```
[28]: original_dir = pathlib.Path('C:\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\train')
new_base_dir = pathlib.Path('C:
    ↪\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\cats_vs_dogs_hw2_p3_tuning_scratch_FINAL')

def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = new_base_dir / subset_name / category
```

```

    os.makedirs(dir)
    fnames = [f"{category}.{i}.jpg"
               for i in range(start_index, end_index)]
    for fname in fnames:
        shutil.copyfile(src=original_dir / fname,
                        dst=dir / fname)
make_subset("train", start_index=0, end_index=4000)          ###4000  ␣
↪###      4000 8000
make_subset("validation", start_index=4000, end_index=6000)  ###2000
make_subset("test", start_index=6000, end_index=7000)        ###1000 TEST

```

```

[29]: from tensorflow import keras
      from tensorflow.keras import layers

```

```

[30]: from tensorflow.keras.utils import image_dataset_from_directory

```

```

train_dataset = image_dataset_from_directory(
    new_base_dir / "train",
    image_size=(180, 180),
    batch_size=32)
validation_dataset = image_dataset_from_directory(
    new_base_dir / "validation",
    image_size=(180, 180),
    batch_size=32)
test_dataset = image_dataset_from_directory(
    new_base_dir / "test",
    image_size=(180, 180),
    batch_size=32)

```

Found 8000 files belonging to 2 classes.
 Found 4000 files belonging to 2 classes.
 Found 2000 files belonging to 2 classes.

```

[31]: import numpy as np
      import tensorflow as tf
      random_numbers = np.random.normal(size=(1000, 16))
      dataset = tf.data.Dataset.from_tensor_slices(random_numbers)

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

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

```



```

        break
reshaped_dataset = dataset.map(lambda x: tf.reshape(x, (4, 4)))
for i, element in enumerate(reshaped_dataset):
    print(element.shape)
    if i >= 2:
        break
#Displaying the shapes of the data and labels yielded by the Dataset
for data_batch, labels_batch in train_dataset:
    print("data batch shape:", data_batch.shape)
    print("labels batch shape:", labels_batch.shape)
    break

```

```

(16,)
(16,)
(16,)
(32, 16)
(32, 16)
(32, 16)
(4, 4)
(4, 4)
(4, 4)
data batch shape: (32, 180, 180, 3)
labels batch shape: (32,)

```

```

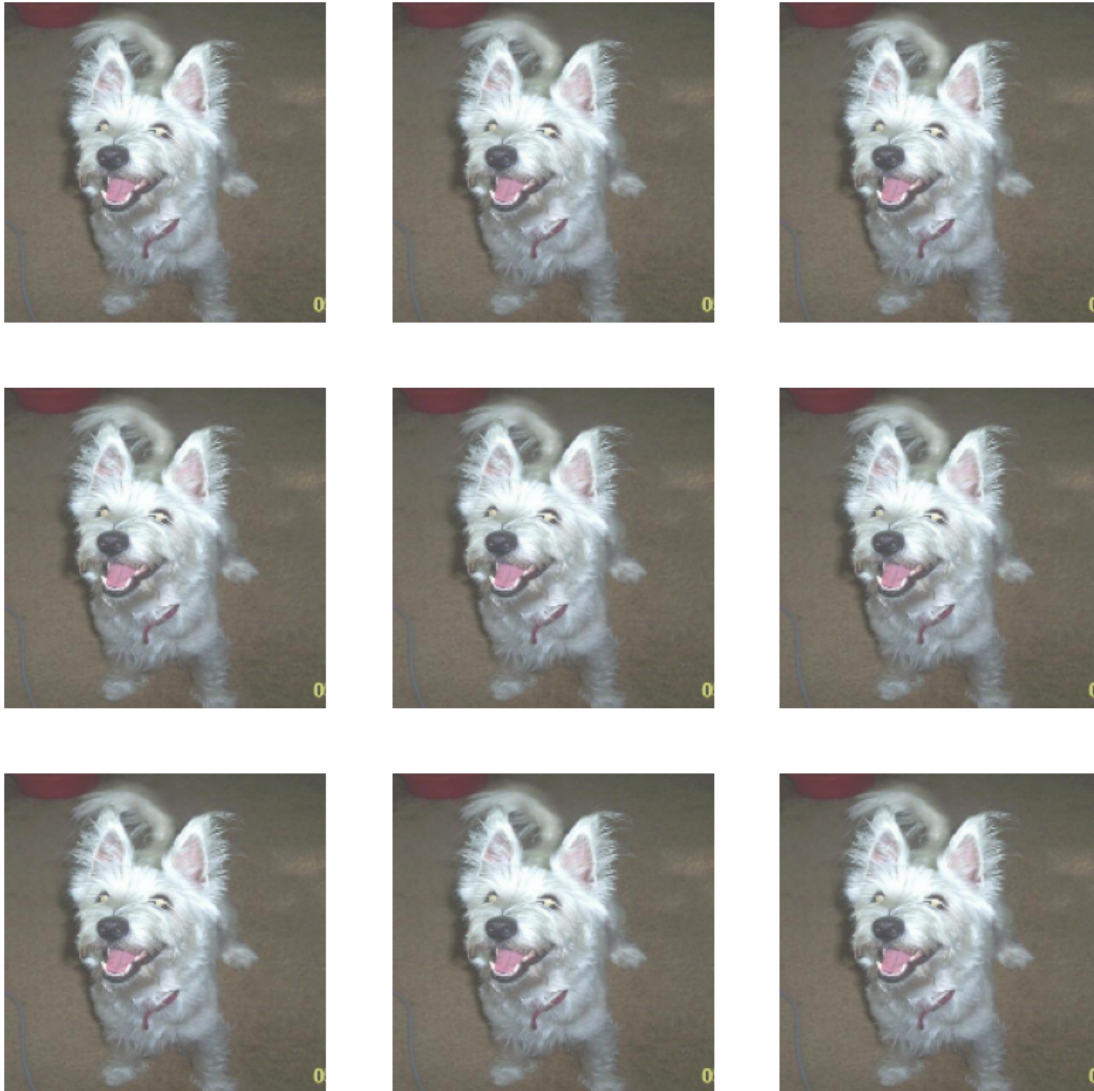
[50]: data_augmentation = keras.Sequential(
    [
        layers.RandomFlip("horizontal"),
        layers.RandomRotation(0.15),
        layers.RandomZoom(0.25),
    ]
)

```

```

[51]: plt.figure(figsize=(10, 10))
for images, _ in train_dataset.take(1):
    for i in range(9):
        augmented_images = data_augmentation(images)
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(augmented_images[0].numpy().astype("uint8"))
        plt.axis("off")

```



```
[52]: # model buildup
inputs = keras.Input(shape=(180, 180, 3))
x = data_augmentation(inputs)
x = layers.Rescaling(1./255)(x)
x = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
```

```

x = layers.Flatten()(x)
x = layers.Dropout(0.4)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs=inputs, outputs=outputs)

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

```

```

[54]: # Training the regularized convnet
callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="convnet_from_scratch_with_augmentation_regularization_hw2.
↳keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=40,
    validation_data=validation_dataset,
    callbacks=callbacks)

```

```

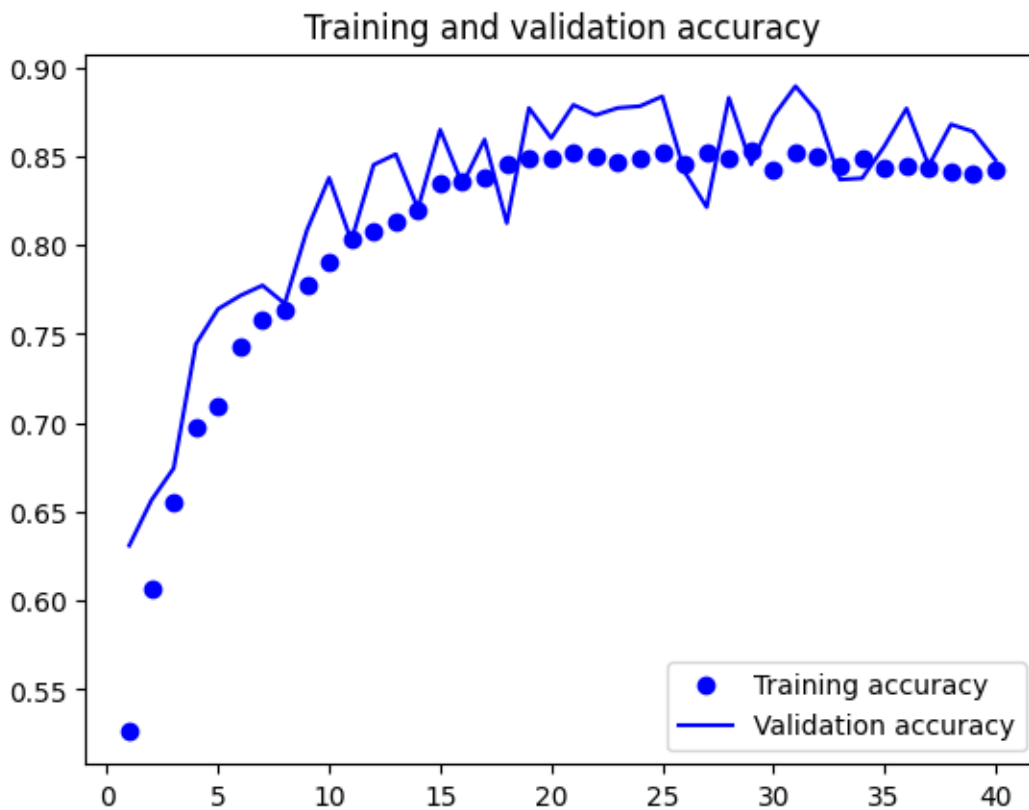
Epoch 1/40
250/250 [=====] - 36s 143ms/step - loss: 0.7048 -
accuracy: 0.5263 - val_loss: 0.6793 - val_accuracy: 0.6310
Epoch 2/40
250/250 [=====] - 35s 142ms/step - loss: 0.6630 -
accuracy: 0.6069 - val_loss: 0.6225 - val_accuracy: 0.6565
Epoch 3/40
250/250 [=====] - 35s 140ms/step - loss: 0.6274 -
accuracy: 0.6554 - val_loss: 0.5959 - val_accuracy: 0.6745
Epoch 4/40
250/250 [=====] - 36s 144ms/step - loss: 0.5883 -
accuracy: 0.6970 - val_loss: 0.5465 - val_accuracy: 0.7442
Epoch 5/40
250/250 [=====] - 36s 144ms/step - loss: 0.5603 -
accuracy: 0.7094 - val_loss: 0.4971 - val_accuracy: 0.7640
Epoch 6/40
250/250 [=====] - 36s 143ms/step - loss: 0.5281 -
accuracy: 0.7431 - val_loss: 0.4754 - val_accuracy: 0.7715
Epoch 7/40
250/250 [=====] - 35s 138ms/step - loss: 0.5113 -
accuracy: 0.7580 - val_loss: 0.4808 - val_accuracy: 0.7772
Epoch 8/40
250/250 [=====] - 35s 140ms/step - loss: 0.4904 -
accuracy: 0.7638 - val_loss: 0.5010 - val_accuracy: 0.7670
Epoch 9/40

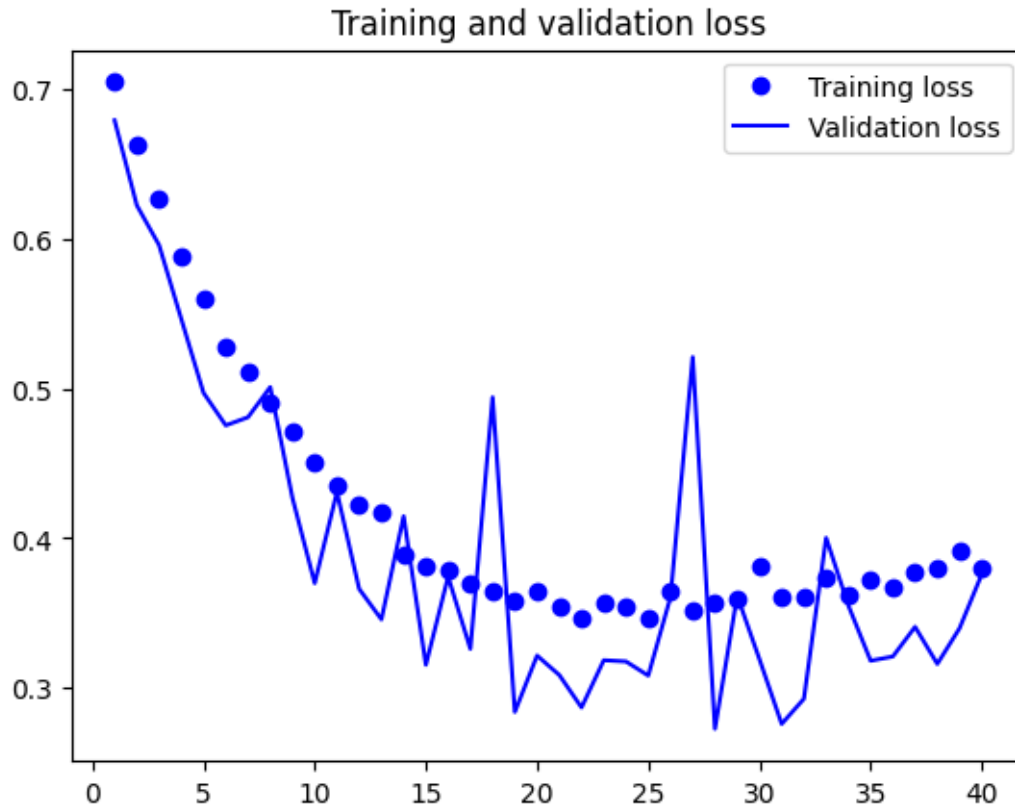
```

250/250 [=====] - 35s 141ms/step - loss: 0.4707 - accuracy: 0.7775 - val_loss: 0.4268 - val_accuracy: 0.8083
Epoch 10/40
250/250 [=====] - 35s 139ms/step - loss: 0.4512 - accuracy: 0.7903 - val_loss: 0.3699 - val_accuracy: 0.8378
Epoch 11/40
250/250 [=====] - 35s 139ms/step - loss: 0.4354 - accuracy: 0.8035 - val_loss: 0.4309 - val_accuracy: 0.8023
Epoch 12/40
250/250 [=====] - 35s 141ms/step - loss: 0.4223 - accuracy: 0.8075 - val_loss: 0.3660 - val_accuracy: 0.8450
Epoch 13/40
250/250 [=====] - 35s 140ms/step - loss: 0.4168 - accuracy: 0.8134 - val_loss: 0.3457 - val_accuracy: 0.8510
Epoch 14/40
250/250 [=====] - 35s 142ms/step - loss: 0.3891 - accuracy: 0.8194 - val_loss: 0.4147 - val_accuracy: 0.8205
Epoch 15/40
250/250 [=====] - 36s 144ms/step - loss: 0.3812 - accuracy: 0.8346 - val_loss: 0.3153 - val_accuracy: 0.8648
Epoch 16/40
250/250 [=====] - 36s 143ms/step - loss: 0.3788 - accuracy: 0.8357 - val_loss: 0.3743 - val_accuracy: 0.8340
Epoch 17/40
250/250 [=====] - 35s 141ms/step - loss: 0.3702 - accuracy: 0.8379 - val_loss: 0.3259 - val_accuracy: 0.8593
Epoch 18/40
250/250 [=====] - 35s 141ms/step - loss: 0.3648 - accuracy: 0.8460 - val_loss: 0.4941 - val_accuracy: 0.8123
Epoch 19/40
250/250 [=====] - 35s 142ms/step - loss: 0.3576 - accuracy: 0.8490 - val_loss: 0.2838 - val_accuracy: 0.8770
Epoch 20/40
250/250 [=====] - 35s 141ms/step - loss: 0.3648 - accuracy: 0.8482 - val_loss: 0.3215 - val_accuracy: 0.8600
Epoch 21/40
250/250 [=====] - 35s 138ms/step - loss: 0.3543 - accuracy: 0.8518 - val_loss: 0.3085 - val_accuracy: 0.8788
Epoch 22/40
250/250 [=====] - 35s 141ms/step - loss: 0.3462 - accuracy: 0.8501 - val_loss: 0.2868 - val_accuracy: 0.8730
Epoch 23/40
250/250 [=====] - 35s 140ms/step - loss: 0.3565 - accuracy: 0.8462 - val_loss: 0.3184 - val_accuracy: 0.8770
Epoch 24/40
250/250 [=====] - 35s 141ms/step - loss: 0.3541 - accuracy: 0.8490 - val_loss: 0.3175 - val_accuracy: 0.8780
Epoch 25/40

250/250 [=====] - 36s 146ms/step - loss: 0.3460 - accuracy: 0.8524 - val_loss: 0.3081 - val_accuracy: 0.8835
Epoch 26/40
250/250 [=====] - 36s 145ms/step - loss: 0.3648 - accuracy: 0.8453 - val_loss: 0.3605 - val_accuracy: 0.8413
Epoch 27/40
250/250 [=====] - 35s 141ms/step - loss: 0.3513 - accuracy: 0.8522 - val_loss: 0.5210 - val_accuracy: 0.8213
Epoch 28/40
250/250 [=====] - 35s 140ms/step - loss: 0.3571 - accuracy: 0.8485 - val_loss: 0.2725 - val_accuracy: 0.8827
Epoch 29/40
250/250 [=====] - 35s 142ms/step - loss: 0.3596 - accuracy: 0.8533 - val_loss: 0.3615 - val_accuracy: 0.8453
Epoch 30/40
250/250 [=====] - 35s 142ms/step - loss: 0.3816 - accuracy: 0.8426 - val_loss: 0.3191 - val_accuracy: 0.8723
Epoch 31/40
250/250 [=====] - 35s 139ms/step - loss: 0.3607 - accuracy: 0.8518 - val_loss: 0.2758 - val_accuracy: 0.8892
Epoch 32/40
250/250 [=====] - 35s 141ms/step - loss: 0.3601 - accuracy: 0.8503 - val_loss: 0.2926 - val_accuracy: 0.8745
Epoch 33/40
250/250 [=====] - 36s 142ms/step - loss: 0.3732 - accuracy: 0.8444 - val_loss: 0.4002 - val_accuracy: 0.8367
Epoch 34/40
250/250 [=====] - 35s 141ms/step - loss: 0.3613 - accuracy: 0.8482 - val_loss: 0.3550 - val_accuracy: 0.8375
Epoch 35/40
250/250 [=====] - 36s 142ms/step - loss: 0.3726 - accuracy: 0.8434 - val_loss: 0.3180 - val_accuracy: 0.8555
Epoch 36/40
250/250 [=====] - 36s 144ms/step - loss: 0.3671 - accuracy: 0.8441 - val_loss: 0.3209 - val_accuracy: 0.8767
Epoch 37/40
250/250 [=====] - 35s 141ms/step - loss: 0.3779 - accuracy: 0.8432 - val_loss: 0.3407 - val_accuracy: 0.8445
Epoch 38/40
250/250 [=====] - 35s 141ms/step - loss: 0.3802 - accuracy: 0.8416 - val_loss: 0.3159 - val_accuracy: 0.8677
Epoch 39/40
250/250 [=====] - 36s 143ms/step - loss: 0.3915 - accuracy: 0.8397 - val_loss: 0.3398 - val_accuracy: 0.8637
Epoch 40/40
250/250 [=====] - 35s 141ms/step - loss: 0.3803 - accuracy: 0.8425 - val_loss: 0.3759 - val_accuracy: 0.8478

```
[55]: # plot the performance
import matplotlib.pyplot as plt
accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
loss = history.history["loss"]
val_loss = history.history["val_loss"]
epochs = range(1, len(accuracy) + 1)
plt.plot(epochs, accuracy, "bo", label="Training accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.legend()
plt.figure()
plt.plot(epochs, loss, "bo", label="Training loss")
plt.plot(epochs, val_loss, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.legend()
plt.show()
```





```
[56]: # Evaluating the model on the test set
```

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

```
63/63 [=====] - 1s 16ms/step - loss: 0.4965 - accuracy: 0.7875
```

```
Test accuracy: 0.788
```

```
[ ]: #cont with part3 but I will increase batch size to 128
```

```
[58]: original_dir = pathlib.Path('C:\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\train')
new_base_dir = pathlib.Path('C:
    ↪\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\cats_vs_dogs_hw2_p3_tuning_scratch_FINAL_v2')

def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = new_base_dir / subset_name / category
```

```

    os.makedirs(dir)
    fnames = [f"{category}.{i}.jpg"
               for i in range(start_index, end_index)]
    for fname in fnames:
        shutil.copyfile(src=original_dir / fname,
                        dst=dir / fname)
make_subset("train", start_index=0, end_index=4000)          ###4000  ␣
↪###      4000 8000
make_subset("validation", start_index=4000, end_index=6000)  ###2000
make_subset("test", start_index=6000, end_index=7000)        ###1000 TEST

```

```

[59]: from tensorflow import keras
      from tensorflow.keras import layers

```

```

[60]: from tensorflow.keras.utils import image_dataset_from_directory

```

```

train_dataset = image_dataset_from_directory(
    new_base_dir / "train",
    image_size=(180, 180),
    batch_size=128)
validation_dataset = image_dataset_from_directory(
    new_base_dir / "validation",
    image_size=(180, 180),
    batch_size=128)
test_dataset = image_dataset_from_directory(
    new_base_dir / "test",
    image_size=(180, 180),
    batch_size=128)

```

Found 8000 files belonging to 2 classes.
Found 4000 files belonging to 2 classes.
Found 2000 files belonging to 2 classes.

```

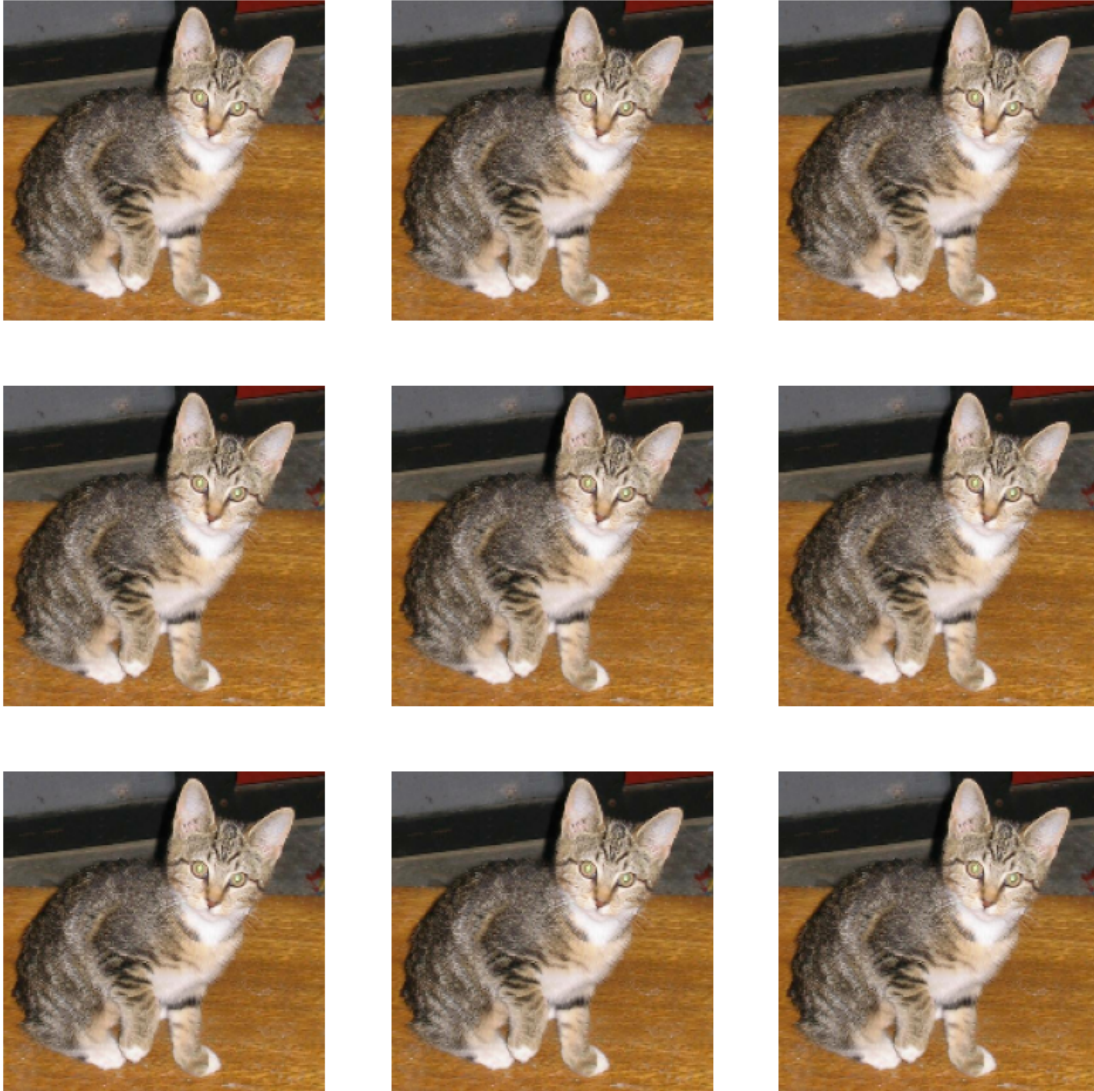
[61]: data_augmentation = keras.Sequential(
    [
        layers.RandomFlip("horizontal"),
        layers.RandomRotation(0.15),
        layers.RandomZoom(0.25),
    ]
)

```

```

[62]: plt.figure(figsize=(10, 10))
      for images, _ in train_dataset.take(1):
          for i in range(9):
              augmented_images = data_augmentation(images)
              ax = plt.subplot(3, 3, i + 1)
              plt.imshow(augmented_images[0].numpy().astype("uint8"))
              plt.axis("off")

```

```
[63]: # model buildup
inputs = keras.Input(shape=(180, 180, 3))
x = data_augmentation(inputs)
x = layers.Rescaling(1./255)(x)
x = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
```

```

x = layers.Flatten()(x)
x = layers.Dropout(0.4)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs=inputs, outputs=outputs)

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

```

```

[64]: # Training the regularized convnet
callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="convnet_from_scratch_with_augmentation_regularization_hw2.
↳keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=40,
    validation_data=validation_dataset,
    callbacks=callbacks)

```

Epoch 1/40

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

63/63 [=====] - 88s 1s/step - loss: 0.7285 - accuracy: 0.5289 - val_loss: 0.6805 - val_accuracy: 0.5130

Epoch 2/40

63/63 [=====] - 77s 1s/step - loss: 0.7038 - accuracy: 0.5715 - val_loss: 0.6717 - val_accuracy: 0.5418

Epoch 3/40

63/63 [=====] - 78s 1s/step - loss: 0.6644 - accuracy: 0.6068 - val_loss: 0.6386 - val_accuracy: 0.6465

Epoch 4/40

63/63 [=====] - 79s 1s/step - loss: 0.6401 - accuracy: 0.6417 - val_loss: 0.6064 - val_accuracy: 0.6585

Epoch 5/40

63/63 [=====] - 78s 1s/step - loss: 0.6170 - accuracy: 0.6565 - val_loss: 0.6187 - val_accuracy: 0.6635

Epoch 6/40

63/63 [=====] - 80s 1s/step - loss: 0.6027 - accuracy: 0.6776 - val_loss: 0.5541 - val_accuracy: 0.7287

Epoch 7/40

63/63 [=====] - 80s 1s/step - loss: 0.5950 - accuracy: 0.6840 - val_loss: 0.5065 - val_accuracy: 0.7585

Epoch 8/40

63/63 [=====] - 77s 1s/step - loss: 0.5663 - accuracy: 0.7131 - val_loss: 0.4943 - val_accuracy: 0.7680

Epoch 9/40

63/63 [=====] - 77s 1s/step - loss: 0.5481 - accuracy: 0.7190 - val_loss: 0.5156 - val_accuracy: 0.7405

Epoch 10/40

63/63 [=====] - 77s 1s/step - loss: 0.5352 - accuracy: 0.7318 - val_loss: 0.4574 - val_accuracy: 0.7872

Epoch 11/40

63/63 [=====] - 76s 1s/step - loss: 0.5150 - accuracy: 0.7485 - val_loss: 0.4750 - val_accuracy: 0.7713

Epoch 12/40
63/63 [=====] - 75s 1s/step - loss: 0.5030 - accuracy:
0.7577 - val_loss: 0.4994 - val_accuracy: 0.7550
Epoch 13/40
63/63 [=====] - 77s 1s/step - loss: 0.4930 - accuracy:
0.7674 - val_loss: 0.4476 - val_accuracy: 0.7910
Epoch 14/40
63/63 [=====] - 77s 1s/step - loss: 0.4818 - accuracy:
0.7715 - val_loss: 0.4305 - val_accuracy: 0.8075
Epoch 15/40
63/63 [=====] - 79s 1s/step - loss: 0.4639 - accuracy:
0.7801 - val_loss: 0.3940 - val_accuracy: 0.8183
Epoch 16/40
63/63 [=====] - 78s 1s/step - loss: 0.4562 - accuracy:
0.7829 - val_loss: 0.5238 - val_accuracy: 0.7495
Epoch 17/40
63/63 [=====] - 79s 1s/step - loss: 0.4517 - accuracy:
0.7900 - val_loss: 0.3949 - val_accuracy: 0.8177
Epoch 18/40
63/63 [=====] - 78s 1s/step - loss: 0.4265 - accuracy:
0.8015 - val_loss: 0.3782 - val_accuracy: 0.8305
Epoch 19/40
63/63 [=====] - 79s 1s/step - loss: 0.4109 - accuracy:
0.8146 - val_loss: 0.3573 - val_accuracy: 0.8393
Epoch 20/40
63/63 [=====] - 79s 1s/step - loss: 0.3921 - accuracy:
0.8231 - val_loss: 0.4462 - val_accuracy: 0.7850
Epoch 21/40
63/63 [=====] - 78s 1s/step - loss: 0.3799 - accuracy:
0.8307 - val_loss: 0.3713 - val_accuracy: 0.8407
Epoch 22/40
63/63 [=====] - 79s 1s/step - loss: 0.3771 - accuracy:
0.8294 - val_loss: 0.3572 - val_accuracy: 0.8535
Epoch 23/40
63/63 [=====] - 76s 1s/step - loss: 0.3502 - accuracy:
0.8462 - val_loss: 0.5005 - val_accuracy: 0.7890
Epoch 24/40
63/63 [=====] - 77s 1s/step - loss: 0.3599 - accuracy:
0.8384 - val_loss: 0.3414 - val_accuracy: 0.8493
Epoch 25/40
63/63 [=====] - 78s 1s/step - loss: 0.3417 - accuracy:
0.8504 - val_loss: 0.2900 - val_accuracy: 0.8715
Epoch 26/40
63/63 [=====] - 78s 1s/step - loss: 0.3338 - accuracy:
0.8546 - val_loss: 0.2865 - val_accuracy: 0.8720
Epoch 27/40
63/63 [=====] - 78s 1s/step - loss: 0.3269 - accuracy:
0.8600 - val_loss: 0.3656 - val_accuracy: 0.8380

```

Epoch 28/40
63/63 [=====] - 76s 1s/step - loss: 0.3224 - accuracy:
0.8558 - val_loss: 0.2957 - val_accuracy: 0.8727
Epoch 29/40
63/63 [=====] - 79s 1s/step - loss: 0.3097 - accuracy:
0.8659 - val_loss: 0.3151 - val_accuracy: 0.8585
Epoch 30/40
63/63 [=====] - 77s 1s/step - loss: 0.2969 - accuracy:
0.8712 - val_loss: 0.3520 - val_accuracy: 0.8622
Epoch 31/40
63/63 [=====] - 77s 1s/step - loss: 0.3012 - accuracy:
0.8724 - val_loss: 0.2957 - val_accuracy: 0.8708
Epoch 32/40
63/63 [=====] - 79s 1s/step - loss: 0.2831 - accuracy:
0.8755 - val_loss: 0.2952 - val_accuracy: 0.8695
Epoch 33/40
63/63 [=====] - 77s 1s/step - loss: 0.2880 - accuracy:
0.8790 - val_loss: 0.2946 - val_accuracy: 0.8800
Epoch 34/40
63/63 [=====] - 77s 1s/step - loss: 0.2592 - accuracy:
0.8907 - val_loss: 0.2432 - val_accuracy: 0.8980
Epoch 35/40
63/63 [=====] - 76s 1s/step - loss: 0.2567 - accuracy:
0.8932 - val_loss: 0.2575 - val_accuracy: 0.8903
Epoch 36/40
63/63 [=====] - 77s 1s/step - loss: 0.2430 - accuracy:
0.9006 - val_loss: 0.3884 - val_accuracy: 0.8457
Epoch 37/40
63/63 [=====] - 78s 1s/step - loss: 0.2476 - accuracy:
0.8960 - val_loss: 0.3018 - val_accuracy: 0.8890
Epoch 38/40
63/63 [=====] - 78s 1s/step - loss: 0.2443 - accuracy:
0.8991 - val_loss: 0.3140 - val_accuracy: 0.8873
Epoch 39/40
63/63 [=====] - 78s 1s/step - loss: 0.2277 - accuracy:
0.9079 - val_loss: 0.2759 - val_accuracy: 0.8870
Epoch 40/40
63/63 [=====] - 74s 1s/step - loss: 0.2368 - accuracy:
0.8988 - val_loss: 0.2924 - val_accuracy: 0.8932

```

```

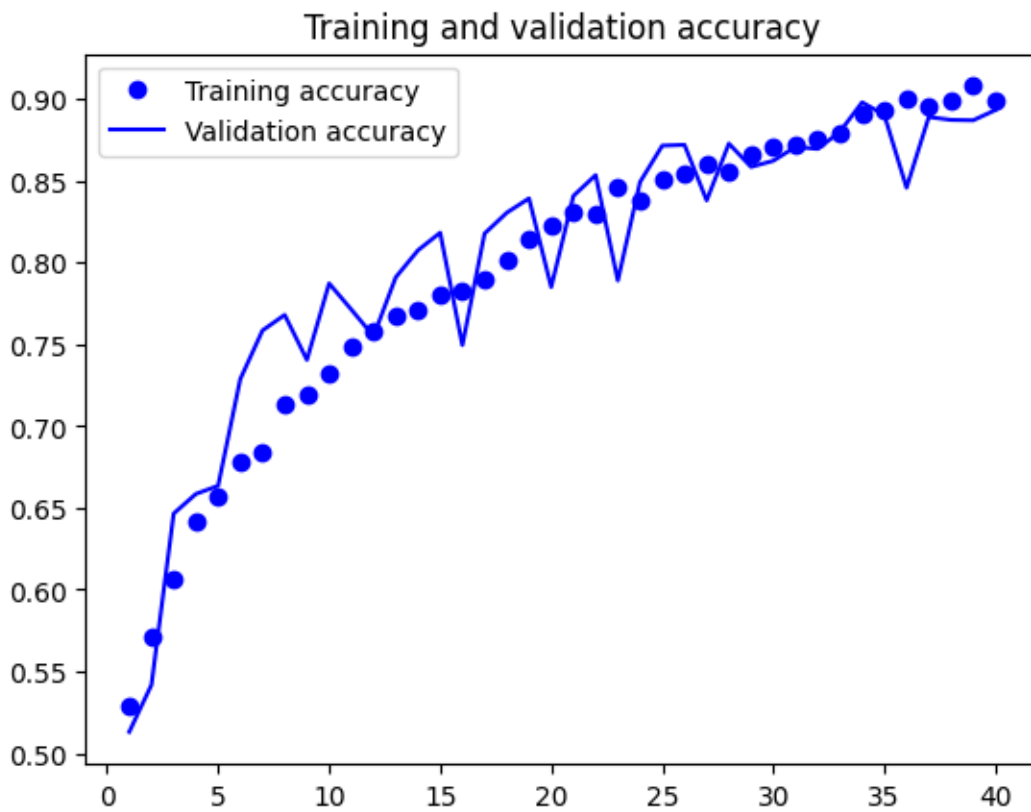
[65]: # plot the performance
import matplotlib.pyplot as plt
accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
loss = history.history["loss"]
val_loss = history.history["val_loss"]
epochs = range(1, len(accuracy) + 1)

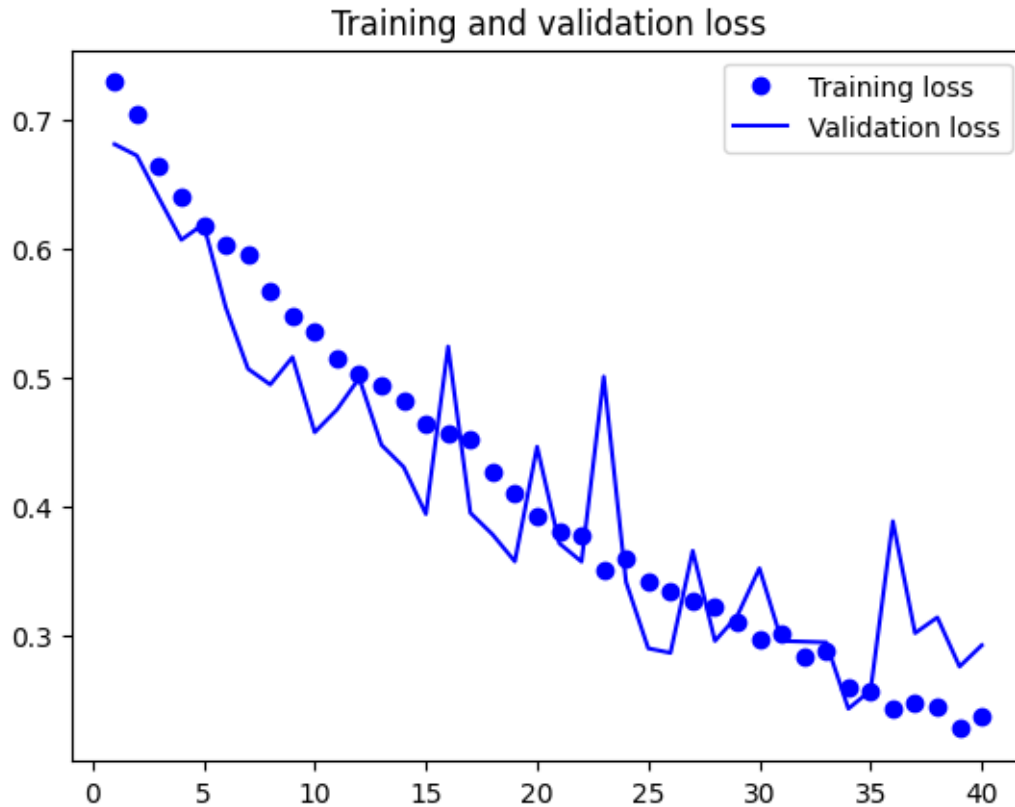
```

```

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

```





```
[66]: # Evaluating the model on the test set
```

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

```
16/16 [=====] - 2s 76ms/step - loss: 0.4965 - accuracy: 0.7875
```

```
Test accuracy: 0.788
```

```
[ ]: # create the model without data_augmentation to compare with my first model
```

```
[67]: original_dir = pathlib.Path('C:\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\train')
new_base_dir = pathlib.Path('C:
    ↪\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\cats_vs_dogs_hw2_noaug')

def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = new_base_dir / subset_name / category
```

```

        os.makedirs(dir)
        fnames = [f"{category}.{i}.jpg"
                   for i in range(start_index, end_index)]
        for fname in fnames:
            shutil.copyfile(src=original_dir / fname,
                            dst=dir / fname)
make_subset("train", start_index=0, end_index=1000)          ###1000  ┘
↪###      1000  2000
make_subset("validation", start_index=1000, end_index=1500)  ###500
make_subset("test", start_index=1500, end_index=2000)        ###500 TEST

```

```
[68]: from tensorflow import keras
      from tensorflow.keras import layers
```

```
[69]: from tensorflow.keras.utils import image_dataset_from_directory
```

```

train_dataset = image_dataset_from_directory(
    new_base_dir / "train",
    image_size=(180, 180),
    batch_size=32)
validation_dataset = image_dataset_from_directory(
    new_base_dir / "validation",
    image_size=(180, 180),
    batch_size=32)
test_dataset = image_dataset_from_directory(
    new_base_dir / "test",
    image_size=(180, 180),
    batch_size=32)

```

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

```
[70]: # model buildup
inputs = keras.Input(shape=(180, 180, 3))
x = data_augmentation(inputs)
x = layers.Rescaling(1./255)(x)
x = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
x = layers.Flatten()(x)

```



```

x = layers.Dropout(0.4)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs=inputs, outputs=outputs)

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

```

```

[71]: # Training the regularized convnet
callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="convnet_from_scratch_with_augmentation_regularization_hw2.
↳keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=150,
    validation_data=validation_dataset,
    callbacks=callbacks)

```

Epoch 1/150

```

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no

```

```

registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
63/63 [=====] - 12s 156ms/step - loss: 0.7796 -
accuracy: 0.5115 - val_loss: 0.6968 - val_accuracy: 0.5000
Epoch 2/150
63/63 [=====] - 9s 142ms/step - loss: 0.7349 -
accuracy: 0.5235 - val_loss: 0.6856 - val_accuracy: 0.6160
Epoch 3/150
63/63 [=====] - 9s 141ms/step - loss: 0.7039 -
accuracy: 0.5440 - val_loss: 0.6757 - val_accuracy: 0.6140
Epoch 4/150
63/63 [=====] - 9s 149ms/step - loss: 0.6930 -
accuracy: 0.5620 - val_loss: 0.6472 - val_accuracy: 0.6090
Epoch 5/150
63/63 [=====] - 9s 143ms/step - loss: 0.6822 -
accuracy: 0.6150 - val_loss: 0.6764 - val_accuracy: 0.5370
Epoch 6/150
63/63 [=====] - 9s 146ms/step - loss: 0.6572 -
accuracy: 0.6100 - val_loss: 0.6468 - val_accuracy: 0.6030
Epoch 7/150
63/63 [=====] - 9s 145ms/step - loss: 0.6368 -
accuracy: 0.6495 - val_loss: 0.9458 - val_accuracy: 0.5150
Epoch 8/150
63/63 [=====] - 9s 144ms/step - loss: 0.6181 -
accuracy: 0.6625 - val_loss: 0.7388 - val_accuracy: 0.5670
Epoch 9/150
63/63 [=====] - 9s 148ms/step - loss: 0.6489 -
accuracy: 0.6690 - val_loss: 0.6094 - val_accuracy: 0.6760
Epoch 10/150
63/63 [=====] - 9s 145ms/step - loss: 0.6002 -
accuracy: 0.6705 - val_loss: 0.6352 - val_accuracy: 0.6610
Epoch 11/150
63/63 [=====] - 9s 144ms/step - loss: 0.5998 -
accuracy: 0.6765 - val_loss: 0.5759 - val_accuracy: 0.7000
Epoch 12/150

```

63/63 [=====] - 9s 142ms/step - loss: 0.5932 -
accuracy: 0.6900 - val_loss: 0.5871 - val_accuracy: 0.7060
Epoch 13/150
63/63 [=====] - 9s 144ms/step - loss: 0.5719 -
accuracy: 0.7095 - val_loss: 0.5524 - val_accuracy: 0.7090
Epoch 14/150
63/63 [=====] - 9s 142ms/step - loss: 0.5707 -
accuracy: 0.6940 - val_loss: 0.6904 - val_accuracy: 0.6340
Epoch 15/150
63/63 [=====] - 9s 142ms/step - loss: 0.5658 -
accuracy: 0.7075 - val_loss: 0.5850 - val_accuracy: 0.7000
Epoch 16/150
63/63 [=====] - 9s 143ms/step - loss: 0.5545 -
accuracy: 0.7205 - val_loss: 0.5302 - val_accuracy: 0.7410
Epoch 17/150
63/63 [=====] - 9s 142ms/step - loss: 0.5359 -
accuracy: 0.7360 - val_loss: 0.6440 - val_accuracy: 0.7330
Epoch 18/150
63/63 [=====] - 9s 145ms/step - loss: 0.5058 -
accuracy: 0.7570 - val_loss: 0.6141 - val_accuracy: 0.7500
Epoch 19/150
63/63 [=====] - 9s 145ms/step - loss: 0.5310 -
accuracy: 0.7450 - val_loss: 0.5305 - val_accuracy: 0.7340
Epoch 20/150
63/63 [=====] - 9s 144ms/step - loss: 0.5115 -
accuracy: 0.7500 - val_loss: 0.5779 - val_accuracy: 0.7250
Epoch 21/150
63/63 [=====] - 9s 144ms/step - loss: 0.5039 -
accuracy: 0.7690 - val_loss: 0.5243 - val_accuracy: 0.7440
Epoch 22/150
63/63 [=====] - 9s 145ms/step - loss: 0.4924 -
accuracy: 0.7600 - val_loss: 0.5111 - val_accuracy: 0.7560
Epoch 23/150
63/63 [=====] - 9s 147ms/step - loss: 0.4971 -
accuracy: 0.7745 - val_loss: 0.5561 - val_accuracy: 0.7360
Epoch 24/150
63/63 [=====] - 9s 147ms/step - loss: 0.4735 -
accuracy: 0.7805 - val_loss: 0.4845 - val_accuracy: 0.7580
Epoch 25/150
63/63 [=====] - 9s 149ms/step - loss: 0.4577 -
accuracy: 0.7950 - val_loss: 0.5870 - val_accuracy: 0.7380
Epoch 26/150
63/63 [=====] - 9s 144ms/step - loss: 0.4636 -
accuracy: 0.7865 - val_loss: 0.5494 - val_accuracy: 0.7780
Epoch 27/150
63/63 [=====] - 9s 144ms/step - loss: 0.4577 -
accuracy: 0.7895 - val_loss: 0.5059 - val_accuracy: 0.7770
Epoch 28/150

63/63 [=====] - 9s 142ms/step - loss: 0.4495 - accuracy: 0.7935 - val_loss: 0.5855 - val_accuracy: 0.7570
Epoch 29/150
63/63 [=====] - 9s 143ms/step - loss: 0.4355 - accuracy: 0.8025 - val_loss: 0.5206 - val_accuracy: 0.7900
Epoch 30/150
63/63 [=====] - 9s 143ms/step - loss: 0.4281 - accuracy: 0.8075 - val_loss: 0.5808 - val_accuracy: 0.7450
Epoch 31/150
63/63 [=====] - 9s 142ms/step - loss: 0.4250 - accuracy: 0.8010 - val_loss: 0.5693 - val_accuracy: 0.7750
Epoch 32/150
63/63 [=====] - 9s 142ms/step - loss: 0.4249 - accuracy: 0.8095 - val_loss: 0.5412 - val_accuracy: 0.7510
Epoch 33/150
63/63 [=====] - 9s 142ms/step - loss: 0.4100 - accuracy: 0.8150 - val_loss: 0.5199 - val_accuracy: 0.7530
Epoch 34/150
63/63 [=====] - 9s 142ms/step - loss: 0.3959 - accuracy: 0.8250 - val_loss: 0.5112 - val_accuracy: 0.7680
Epoch 35/150
63/63 [=====] - 9s 141ms/step - loss: 0.3975 - accuracy: 0.8245 - val_loss: 0.5043 - val_accuracy: 0.7640
Epoch 36/150
63/63 [=====] - 9s 142ms/step - loss: 0.4045 - accuracy: 0.8270 - val_loss: 0.5321 - val_accuracy: 0.7750
Epoch 37/150
63/63 [=====] - 9s 142ms/step - loss: 0.3639 - accuracy: 0.8380 - val_loss: 0.5956 - val_accuracy: 0.7880
Epoch 38/150
63/63 [=====] - 9s 141ms/step - loss: 0.3752 - accuracy: 0.8315 - val_loss: 0.5887 - val_accuracy: 0.8020
Epoch 39/150
63/63 [=====] - 9s 144ms/step - loss: 0.3473 - accuracy: 0.8415 - val_loss: 0.5854 - val_accuracy: 0.7480
Epoch 40/150
63/63 [=====] - 9s 141ms/step - loss: 0.3512 - accuracy: 0.8490 - val_loss: 0.5056 - val_accuracy: 0.7840
Epoch 41/150
63/63 [=====] - 9s 141ms/step - loss: 0.3527 - accuracy: 0.8505 - val_loss: 0.5483 - val_accuracy: 0.7980
Epoch 42/150
63/63 [=====] - 9s 148ms/step - loss: 0.3461 - accuracy: 0.8505 - val_loss: 0.5986 - val_accuracy: 0.7540
Epoch 43/150
63/63 [=====] - 9s 142ms/step - loss: 0.3544 - accuracy: 0.8515 - val_loss: 0.8841 - val_accuracy: 0.7340
Epoch 44/150

63/63 [=====] - 9s 146ms/step - loss: 0.3455 -
accuracy: 0.8530 - val_loss: 0.4843 - val_accuracy: 0.8050
Epoch 45/150
63/63 [=====] - 9s 143ms/step - loss: 0.3308 -
accuracy: 0.8510 - val_loss: 0.4902 - val_accuracy: 0.7910
Epoch 46/150
63/63 [=====] - 9s 147ms/step - loss: 0.3135 -
accuracy: 0.8650 - val_loss: 0.5509 - val_accuracy: 0.7940
Epoch 47/150
63/63 [=====] - 9s 147ms/step - loss: 0.3316 -
accuracy: 0.8560 - val_loss: 0.5158 - val_accuracy: 0.7580
Epoch 48/150
63/63 [=====] - 9s 145ms/step - loss: 0.3112 -
accuracy: 0.8750 - val_loss: 0.7120 - val_accuracy: 0.7620
Epoch 49/150
63/63 [=====] - 9s 142ms/step - loss: 0.3184 -
accuracy: 0.8685 - val_loss: 0.5213 - val_accuracy: 0.8030
Epoch 50/150
63/63 [=====] - 9s 148ms/step - loss: 0.3224 -
accuracy: 0.8665 - val_loss: 0.5734 - val_accuracy: 0.7750
Epoch 51/150
63/63 [=====] - 10s 150ms/step - loss: 0.2863 -
accuracy: 0.8840 - val_loss: 0.5929 - val_accuracy: 0.7910
Epoch 52/150
63/63 [=====] - 9s 148ms/step - loss: 0.3018 -
accuracy: 0.8720 - val_loss: 0.6292 - val_accuracy: 0.7920
Epoch 53/150
63/63 [=====] - 9s 149ms/step - loss: 0.3019 -
accuracy: 0.8735 - val_loss: 0.6137 - val_accuracy: 0.8010
Epoch 54/150
63/63 [=====] - 9s 148ms/step - loss: 0.2961 -
accuracy: 0.8790 - val_loss: 0.5618 - val_accuracy: 0.8140
Epoch 55/150
63/63 [=====] - 9s 148ms/step - loss: 0.2639 -
accuracy: 0.8920 - val_loss: 0.8395 - val_accuracy: 0.7630
Epoch 56/150
63/63 [=====] - 9s 148ms/step - loss: 0.2745 -
accuracy: 0.8885 - val_loss: 0.5007 - val_accuracy: 0.7950
Epoch 57/150
63/63 [=====] - 9s 148ms/step - loss: 0.2720 -
accuracy: 0.8850 - val_loss: 0.5344 - val_accuracy: 0.7940
Epoch 58/150
63/63 [=====] - 9s 148ms/step - loss: 0.2854 -
accuracy: 0.8880 - val_loss: 0.9281 - val_accuracy: 0.7190
Epoch 59/150
63/63 [=====] - 9s 148ms/step - loss: 0.2717 -
accuracy: 0.8825 - val_loss: 0.5456 - val_accuracy: 0.8020
Epoch 60/150

63/63 [=====] - 9s 146ms/step - loss: 0.2436 -
accuracy: 0.9050 - val_loss: 0.5398 - val_accuracy: 0.7960
Epoch 61/150
63/63 [=====] - 9s 148ms/step - loss: 0.2486 -
accuracy: 0.8975 - val_loss: 0.8358 - val_accuracy: 0.7700
Epoch 62/150
63/63 [=====] - 9s 147ms/step - loss: 0.2324 -
accuracy: 0.9040 - val_loss: 0.6440 - val_accuracy: 0.8040
Epoch 63/150
63/63 [=====] - 9s 146ms/step - loss: 0.2549 -
accuracy: 0.8955 - val_loss: 0.6888 - val_accuracy: 0.7690
Epoch 64/150
63/63 [=====] - 9s 142ms/step - loss: 0.2606 -
accuracy: 0.8940 - val_loss: 0.5880 - val_accuracy: 0.7940
Epoch 65/150
63/63 [=====] - 9s 143ms/step - loss: 0.2443 -
accuracy: 0.9020 - val_loss: 0.6782 - val_accuracy: 0.7860
Epoch 66/150
63/63 [=====] - 9s 144ms/step - loss: 0.2541 -
accuracy: 0.9035 - val_loss: 0.6587 - val_accuracy: 0.7810
Epoch 67/150
63/63 [=====] - 9s 145ms/step - loss: 0.2486 -
accuracy: 0.9085 - val_loss: 0.9676 - val_accuracy: 0.7630
Epoch 68/150
63/63 [=====] - 9s 144ms/step - loss: 0.2587 -
accuracy: 0.8960 - val_loss: 0.5608 - val_accuracy: 0.8200
Epoch 69/150
63/63 [=====] - 9s 148ms/step - loss: 0.2396 -
accuracy: 0.9040 - val_loss: 0.6002 - val_accuracy: 0.7970
Epoch 70/150
63/63 [=====] - 9s 147ms/step - loss: 0.2551 -
accuracy: 0.8995 - val_loss: 0.6335 - val_accuracy: 0.8020
Epoch 71/150
63/63 [=====] - 9s 142ms/step - loss: 0.2146 -
accuracy: 0.9155 - val_loss: 0.8194 - val_accuracy: 0.8080
Epoch 72/150
63/63 [=====] - 9s 143ms/step - loss: 0.2359 -
accuracy: 0.9125 - val_loss: 0.5696 - val_accuracy: 0.8100
Epoch 73/150
63/63 [=====] - 9s 146ms/step - loss: 0.2288 -
accuracy: 0.9180 - val_loss: 0.7201 - val_accuracy: 0.8100
Epoch 74/150
63/63 [=====] - 9s 147ms/step - loss: 0.2285 -
accuracy: 0.9085 - val_loss: 0.7506 - val_accuracy: 0.8040
Epoch 75/150
63/63 [=====] - 9s 144ms/step - loss: 0.2174 -
accuracy: 0.9095 - val_loss: 0.7407 - val_accuracy: 0.7970
Epoch 76/150

63/63 [=====] - 9s 147ms/step - loss: 0.2068 -
accuracy: 0.9200 - val_loss: 0.8218 - val_accuracy: 0.7660
Epoch 77/150
63/63 [=====] - 9s 144ms/step - loss: 0.2205 -
accuracy: 0.9125 - val_loss: 0.7263 - val_accuracy: 0.8050
Epoch 78/150
63/63 [=====] - 9s 147ms/step - loss: 0.2524 -
accuracy: 0.9120 - val_loss: 0.8976 - val_accuracy: 0.7750
Epoch 79/150
63/63 [=====] - 9s 142ms/step - loss: 0.2126 -
accuracy: 0.9130 - val_loss: 0.7739 - val_accuracy: 0.7890
Epoch 80/150
63/63 [=====] - 10s 154ms/step - loss: 0.2285 -
accuracy: 0.9170 - val_loss: 0.8458 - val_accuracy: 0.7930
Epoch 81/150
63/63 [=====] - 10s 152ms/step - loss: 0.2227 -
accuracy: 0.9170 - val_loss: 0.6856 - val_accuracy: 0.7960
Epoch 82/150
63/63 [=====] - 9s 144ms/step - loss: 0.2173 -
accuracy: 0.9180 - val_loss: 0.8016 - val_accuracy: 0.7750
Epoch 83/150
63/63 [=====] - 9s 149ms/step - loss: 0.2004 -
accuracy: 0.9230 - val_loss: 0.7335 - val_accuracy: 0.8110
Epoch 84/150
63/63 [=====] - 9s 149ms/step - loss: 0.2297 -
accuracy: 0.9160 - val_loss: 1.0161 - val_accuracy: 0.7380
Epoch 85/150
63/63 [=====] - 9s 148ms/step - loss: 0.1951 -
accuracy: 0.9285 - val_loss: 0.6432 - val_accuracy: 0.8040
Epoch 86/150
63/63 [=====] - 10s 151ms/step - loss: 0.1954 -
accuracy: 0.9230 - val_loss: 0.8657 - val_accuracy: 0.7920
Epoch 87/150
63/63 [=====] - 10s 155ms/step - loss: 0.1752 -
accuracy: 0.9420 - val_loss: 1.1690 - val_accuracy: 0.7720
Epoch 88/150
63/63 [=====] - 10s 152ms/step - loss: 0.2217 -
accuracy: 0.9180 - val_loss: 0.8485 - val_accuracy: 0.8200
Epoch 89/150
63/63 [=====] - 10s 153ms/step - loss: 0.2080 -
accuracy: 0.9225 - val_loss: 0.7481 - val_accuracy: 0.8020
Epoch 90/150
63/63 [=====] - 10s 151ms/step - loss: 0.2245 -
accuracy: 0.9230 - val_loss: 0.7882 - val_accuracy: 0.7820
Epoch 91/150
63/63 [=====] - 9s 148ms/step - loss: 0.2331 -
accuracy: 0.9240 - val_loss: 0.6301 - val_accuracy: 0.8070
Epoch 92/150

63/63 [=====] - 9s 146ms/step - loss: 0.2298 -
accuracy: 0.9230 - val_loss: 0.7477 - val_accuracy: 0.8100
Epoch 93/150
63/63 [=====] - 9s 147ms/step - loss: 0.1781 -
accuracy: 0.9330 - val_loss: 0.7426 - val_accuracy: 0.8240
Epoch 94/150
63/63 [=====] - 9s 148ms/step - loss: 0.2134 -
accuracy: 0.9230 - val_loss: 0.7700 - val_accuracy: 0.7870
Epoch 95/150
63/63 [=====] - 9s 147ms/step - loss: 0.2033 -
accuracy: 0.9280 - val_loss: 0.6187 - val_accuracy: 0.8200
Epoch 96/150
63/63 [=====] - 9s 142ms/step - loss: 0.2002 -
accuracy: 0.9305 - val_loss: 0.7037 - val_accuracy: 0.8240
Epoch 97/150
63/63 [=====] - 9s 143ms/step - loss: 0.2151 -
accuracy: 0.9250 - val_loss: 0.7439 - val_accuracy: 0.8080
Epoch 98/150
63/63 [=====] - 9s 146ms/step - loss: 0.1879 -
accuracy: 0.9310 - val_loss: 0.9842 - val_accuracy: 0.8030
Epoch 99/150
63/63 [=====] - 9s 146ms/step - loss: 0.2302 -
accuracy: 0.9230 - val_loss: 0.7847 - val_accuracy: 0.8080
Epoch 100/150
63/63 [=====] - 9s 148ms/step - loss: 0.2171 -
accuracy: 0.9205 - val_loss: 0.7903 - val_accuracy: 0.8290
Epoch 101/150
63/63 [=====] - 9s 148ms/step - loss: 0.2085 -
accuracy: 0.9260 - val_loss: 0.8668 - val_accuracy: 0.8150
Epoch 102/150
63/63 [=====] - 9s 146ms/step - loss: 0.2136 -
accuracy: 0.9330 - val_loss: 0.7176 - val_accuracy: 0.8150
Epoch 103/150
63/63 [=====] - 9s 145ms/step - loss: 0.2061 -
accuracy: 0.9305 - val_loss: 0.7415 - val_accuracy: 0.8180
Epoch 104/150
63/63 [=====] - 9s 145ms/step - loss: 0.1783 -
accuracy: 0.9420 - val_loss: 1.0466 - val_accuracy: 0.8050
Epoch 105/150
63/63 [=====] - 9s 143ms/step - loss: 0.2083 -
accuracy: 0.9260 - val_loss: 0.8668 - val_accuracy: 0.8190
Epoch 106/150
63/63 [=====] - 9s 145ms/step - loss: 0.1944 -
accuracy: 0.9355 - val_loss: 0.8708 - val_accuracy: 0.7960
Epoch 107/150
63/63 [=====] - 9s 145ms/step - loss: 0.1843 -
accuracy: 0.9375 - val_loss: 0.8111 - val_accuracy: 0.8080
Epoch 108/150

63/63 [=====] - 9s 147ms/step - loss: 0.1971 -
accuracy: 0.9355 - val_loss: 0.7062 - val_accuracy: 0.8180
Epoch 109/150
63/63 [=====] - 9s 146ms/step - loss: 0.2140 -
accuracy: 0.9265 - val_loss: 0.9188 - val_accuracy: 0.7920
Epoch 110/150
63/63 [=====] - 9s 146ms/step - loss: 0.2103 -
accuracy: 0.9250 - val_loss: 1.1467 - val_accuracy: 0.7880
Epoch 111/150
63/63 [=====] - 9s 144ms/step - loss: 0.2169 -
accuracy: 0.9210 - val_loss: 0.8432 - val_accuracy: 0.8000
Epoch 112/150
63/63 [=====] - 9s 145ms/step - loss: 0.1804 -
accuracy: 0.9350 - val_loss: 1.2253 - val_accuracy: 0.7870
Epoch 113/150
63/63 [=====] - 9s 146ms/step - loss: 0.1911 -
accuracy: 0.9365 - val_loss: 0.9914 - val_accuracy: 0.8180
Epoch 114/150
63/63 [=====] - 9s 145ms/step - loss: 0.1780 -
accuracy: 0.9430 - val_loss: 1.0184 - val_accuracy: 0.8040
Epoch 115/150
63/63 [=====] - 9s 145ms/step - loss: 0.1924 -
accuracy: 0.9345 - val_loss: 0.9894 - val_accuracy: 0.8300
Epoch 116/150
63/63 [=====] - 9s 143ms/step - loss: 0.2041 -
accuracy: 0.9305 - val_loss: 0.7171 - val_accuracy: 0.8020
Epoch 117/150
63/63 [=====] - 9s 143ms/step - loss: 0.1968 -
accuracy: 0.9355 - val_loss: 1.0654 - val_accuracy: 0.8260
Epoch 118/150
63/63 [=====] - 9s 148ms/step - loss: 0.1687 -
accuracy: 0.9385 - val_loss: 0.7760 - val_accuracy: 0.8300
Epoch 119/150
63/63 [=====] - 10s 155ms/step - loss: 0.2042 -
accuracy: 0.9325 - val_loss: 0.7897 - val_accuracy: 0.8070
Epoch 120/150
63/63 [=====] - 9s 150ms/step - loss: 0.1775 -
accuracy: 0.9410 - val_loss: 1.0494 - val_accuracy: 0.8280
Epoch 121/150
63/63 [=====] - 9s 146ms/step - loss: 0.1923 -
accuracy: 0.9355 - val_loss: 0.8610 - val_accuracy: 0.8170
Epoch 122/150
63/63 [=====] - 9s 146ms/step - loss: 0.1943 -
accuracy: 0.9335 - val_loss: 1.2758 - val_accuracy: 0.7890
Epoch 123/150
63/63 [=====] - 9s 148ms/step - loss: 0.2463 -
accuracy: 0.9270 - val_loss: 0.6924 - val_accuracy: 0.8210
Epoch 124/150

63/63 [=====] - 9s 148ms/step - loss: 0.1799 - accuracy: 0.9420 - val_loss: 1.4488 - val_accuracy: 0.7970
Epoch 125/150
63/63 [=====] - 10s 153ms/step - loss: 0.2169 - accuracy: 0.9245 - val_loss: 0.7361 - val_accuracy: 0.8030
Epoch 126/150
63/63 [=====] - 9s 148ms/step - loss: 0.2004 - accuracy: 0.9380 - val_loss: 1.0756 - val_accuracy: 0.7650
Epoch 127/150
63/63 [=====] - 9s 146ms/step - loss: 0.2157 - accuracy: 0.9350 - val_loss: 1.0452 - val_accuracy: 0.8070
Epoch 128/150
63/63 [=====] - 9s 147ms/step - loss: 0.1915 - accuracy: 0.9390 - val_loss: 1.1154 - val_accuracy: 0.7890
Epoch 129/150
63/63 [=====] - 9s 149ms/step - loss: 0.2148 - accuracy: 0.9285 - val_loss: 0.8023 - val_accuracy: 0.8270
Epoch 130/150
63/63 [=====] - 9s 145ms/step - loss: 0.1677 - accuracy: 0.9420 - val_loss: 1.1780 - val_accuracy: 0.8120
Epoch 131/150
63/63 [=====] - 9s 147ms/step - loss: 0.2162 - accuracy: 0.9245 - val_loss: 0.8518 - val_accuracy: 0.8200
Epoch 132/150
63/63 [=====] - 9s 145ms/step - loss: 0.1970 - accuracy: 0.9355 - val_loss: 0.9291 - val_accuracy: 0.7850
Epoch 133/150
63/63 [=====] - 9s 145ms/step - loss: 0.1805 - accuracy: 0.9405 - val_loss: 1.0939 - val_accuracy: 0.7760
Epoch 134/150
63/63 [=====] - 9s 144ms/step - loss: 0.2072 - accuracy: 0.9340 - val_loss: 0.8310 - val_accuracy: 0.8050
Epoch 135/150
63/63 [=====] - 9s 146ms/step - loss: 0.2178 - accuracy: 0.9350 - val_loss: 0.9364 - val_accuracy: 0.8160
Epoch 136/150
63/63 [=====] - 9s 146ms/step - loss: 0.2363 - accuracy: 0.9310 - val_loss: 1.0462 - val_accuracy: 0.8090
Epoch 137/150
63/63 [=====] - 9s 144ms/step - loss: 0.2052 - accuracy: 0.9365 - val_loss: 0.7999 - val_accuracy: 0.8150
Epoch 138/150
63/63 [=====] - 9s 148ms/step - loss: 0.1943 - accuracy: 0.9430 - val_loss: 0.8391 - val_accuracy: 0.8240
Epoch 139/150
63/63 [=====] - 9s 146ms/step - loss: 0.1723 - accuracy: 0.9395 - val_loss: 1.3928 - val_accuracy: 0.8070
Epoch 140/150

```

63/63 [=====] - 9s 146ms/step - loss: 0.2388 -
accuracy: 0.9395 - val_loss: 1.2592 - val_accuracy: 0.7520
Epoch 141/150
63/63 [=====] - 9s 144ms/step - loss: 0.2163 -
accuracy: 0.9345 - val_loss: 1.1254 - val_accuracy: 0.8290
Epoch 142/150
63/63 [=====] - 9s 148ms/step - loss: 0.1970 -
accuracy: 0.9390 - val_loss: 0.8325 - val_accuracy: 0.8300
Epoch 143/150
63/63 [=====] - 9s 146ms/step - loss: 0.2270 -
accuracy: 0.9320 - val_loss: 0.8097 - val_accuracy: 0.8240
Epoch 144/150
63/63 [=====] - 9s 148ms/step - loss: 0.1904 -
accuracy: 0.9335 - val_loss: 1.0664 - val_accuracy: 0.8290
Epoch 145/150
63/63 [=====] - 9s 145ms/step - loss: 0.2083 -
accuracy: 0.9320 - val_loss: 0.8473 - val_accuracy: 0.8250
Epoch 146/150
63/63 [=====] - 9s 144ms/step - loss: 0.2058 -
accuracy: 0.9345 - val_loss: 1.6110 - val_accuracy: 0.6870
Epoch 147/150
63/63 [=====] - 9s 147ms/step - loss: 0.1921 -
accuracy: 0.9370 - val_loss: 1.4289 - val_accuracy: 0.8120
Epoch 148/150
63/63 [=====] - 9s 143ms/step - loss: 0.2035 -
accuracy: 0.9365 - val_loss: 0.9581 - val_accuracy: 0.8200
Epoch 149/150
63/63 [=====] - 9s 146ms/step - loss: 0.2205 -
accuracy: 0.9285 - val_loss: 0.9538 - val_accuracy: 0.8040
Epoch 150/150
63/63 [=====] - 9s 143ms/step - loss: 0.2247 -
accuracy: 0.9255 - val_loss: 1.7716 - val_accuracy: 0.7390

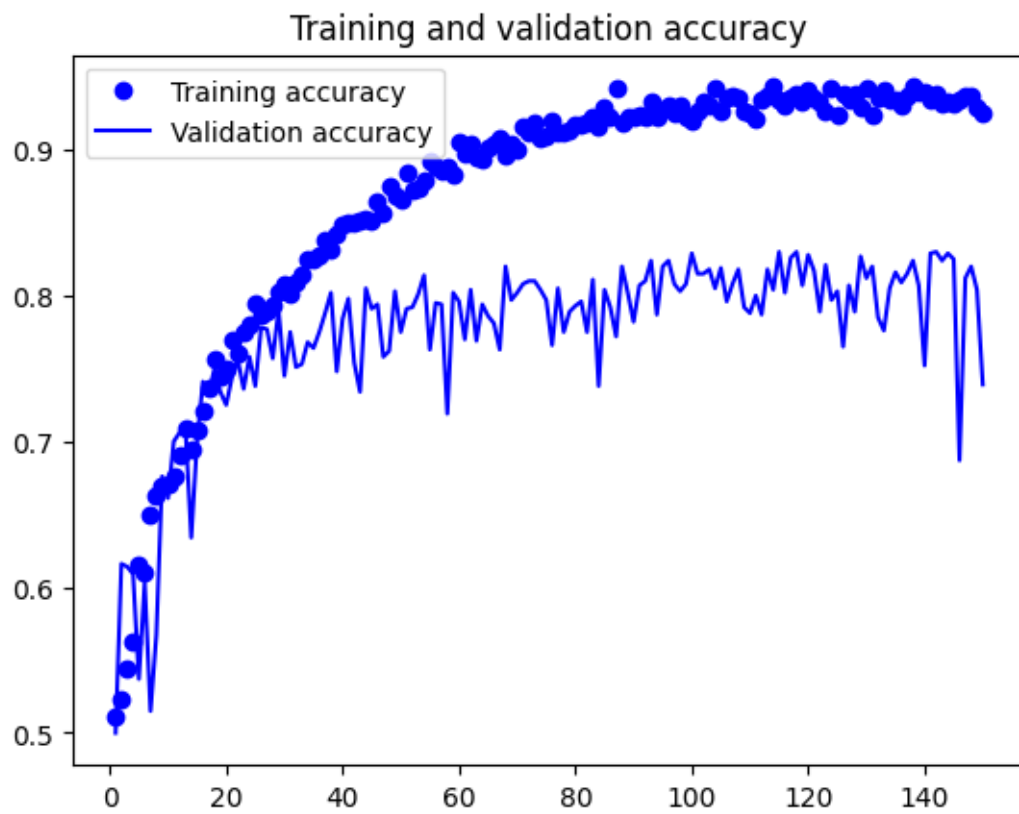
```

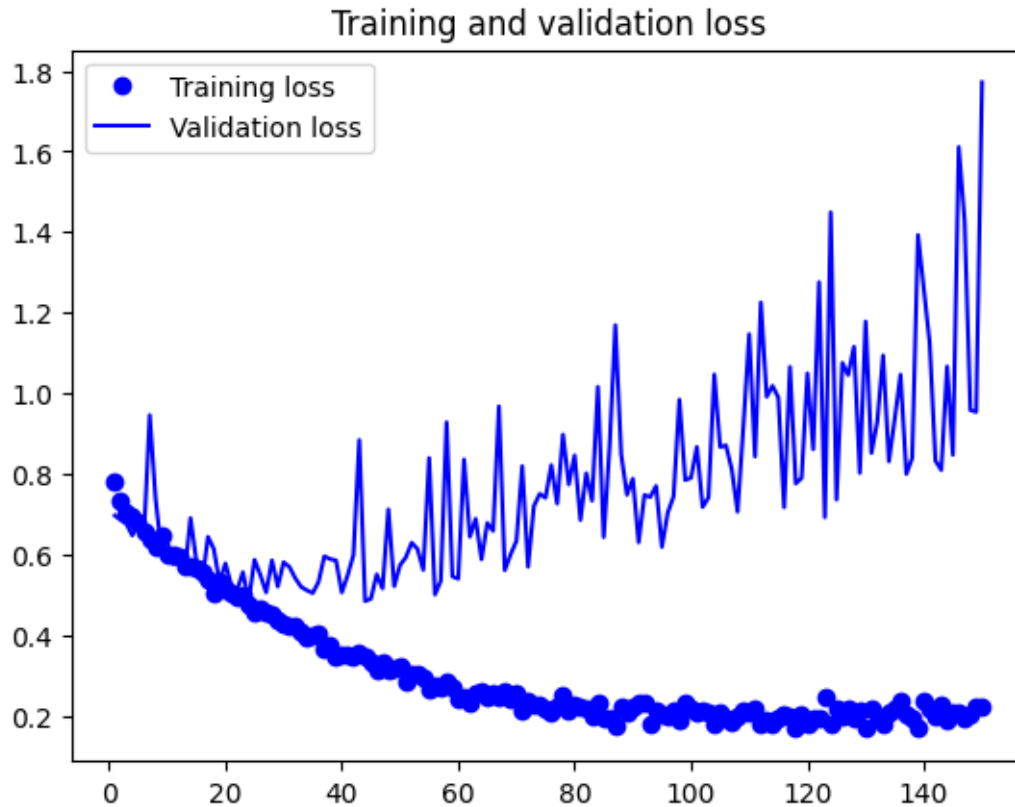
```

[72]: # plot the performance
import matplotlib.pyplot as plt
accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
loss = history.history["loss"]
val_loss = history.history["val_loss"]
epochs = range(1, len(accuracy) + 1)
plt.plot(epochs, accuracy, "bo", label="Training accuracy")
plt.plot(epochs, val_accuracy, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.legend()
plt.figure()
plt.plot(epochs, loss, "bo", label="Training loss")
plt.plot(epochs, val_loss, "b", label="Validation loss")

```

```
plt.title("Training and validation loss")
plt.legend()
plt.show()
```





```
[73]: # Evaluating the model on the test set
```

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

```
32/32 [=====] - 1s 16ms/step - loss: 0.5208 - accuracy: 0.7720
```

```
Test accuracy: 0.772
```

```
[ ]: # create a model with more layers to see if accuracy can be improved above 80%
```

```
[74]: original_dir = pathlib.Path('C:\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\train')
new_base_dir = pathlib.Path('C:
    ↪\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\cats_vs_dogs_hw2_morelayer')

def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = new_base_dir / subset_name / category
```

```

    os.makedirs(dir)
    fnames = [f"{category}.{i}.jpg"
               for i in range(start_index, end_index)]
    for fname in fnames:
        shutil.copyfile(src=original_dir / fname,
                        dst=dir / fname)
make_subset("train", start_index=0, end_index=1000)          ###1000  ␣
↪###      1000  2000
make_subset("validation", start_index=1000, end_index=1500)  ###500
make_subset("test", start_index=1500, end_index=2000)        ###500 TEST

```

[75]: `from tensorflow.keras.utils import image_dataset_from_directory`

```

train_dataset = image_dataset_from_directory(
    new_base_dir / "train",
    image_size=(180, 180),
    batch_size=32)
validation_dataset = image_dataset_from_directory(
    new_base_dir / "validation",
    image_size=(180, 180),
    batch_size=32)
test_dataset = image_dataset_from_directory(
    new_base_dir / "test",
    image_size=(180, 180),
    batch_size=32)

```

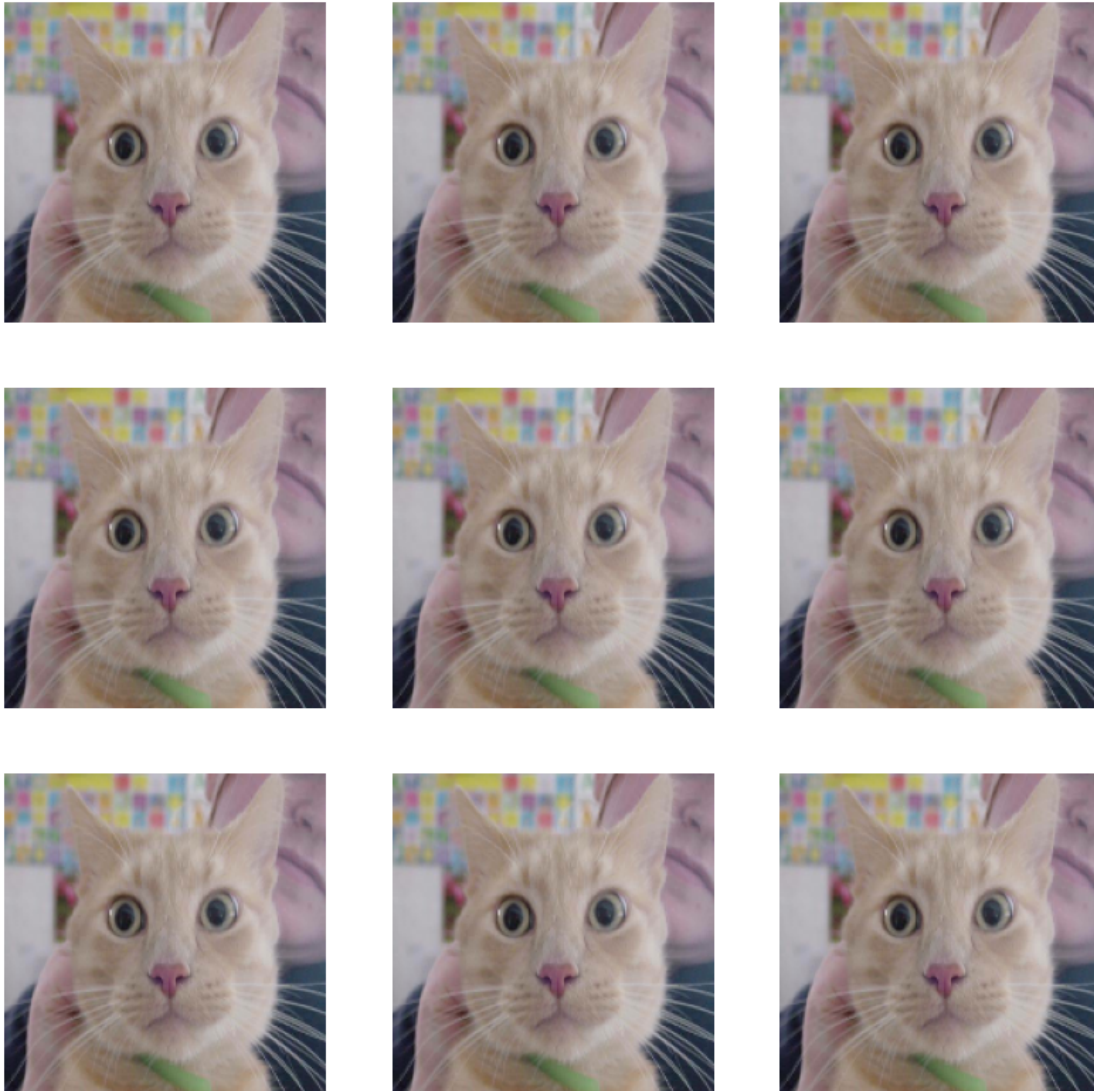
Found 2000 files belonging to 2 classes.

Found 1000 files belonging to 2 classes.

Found 1000 files belonging to 2 classes.

[76]: `data_augmentation = keras.Sequential(`
 `[`
 `layers.RandomFlip("horizontal"),`
 `layers.RandomRotation(0.2),`
 `layers.RandomZoom(0.2),`
 `]`
`)`

[77]: `plt.figure(figsize=(10, 10))`
`for images, _ in train_dataset.take(1):`
 `for i in range(9):`
 `augmented_images = data_augmentation(images)`
 `ax = plt.subplot(3, 3, i + 1)`
 `plt.imshow(augmented_images[0].numpy().astype("uint8"))`
 `plt.axis("off")`



```
[80]: # model buildup, enrich the layers
inputs = keras.Input(shape=(180, 180, 3))
x = data_augmentation(inputs)
x = layers.Rescaling(1./255)(x)
x = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=256, kernel_size=3, activation="relu")(x)
```

```

x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=512, kernel_size=3, activation="relu")(x)
x = layers.Flatten()(x)
x = layers.Dropout(0.4)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs=inputs, outputs=outputs)

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

```

```

[81]: # Training the regularized convnet
callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="convnet_from_scratch_with_augmentation_regularization_hw2.
↳keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=40,
    validation_data=validation_dataset,
    callbacks=callbacks)

```

Epoch 1/40

```

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.

```


WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

63/63 [=====] - 38s 574ms/step - loss: 0.7510 - accuracy: 0.4985 - val_loss: 0.6851 - val_accuracy: 0.5080

Epoch 2/40

63/63 [=====] - 36s 567ms/step - loss: 0.6961 - accuracy: 0.5510 - val_loss: 0.6792 - val_accuracy: 0.6170

Epoch 3/40

63/63 [=====] - 38s 597ms/step - loss: 0.6891 - accuracy: 0.5685 - val_loss: 0.6723 - val_accuracy: 0.6060

Epoch 4/40

63/63 [=====] - 37s 591ms/step - loss: 0.6781 - accuracy: 0.6015 - val_loss: 0.8837 - val_accuracy: 0.5400

Epoch 5/40

63/63 [=====] - 37s 589ms/step - loss: 0.6471 - accuracy: 0.6230 - val_loss: 1.1680 - val_accuracy: 0.5070

Epoch 6/40

63/63 [=====] - 37s 582ms/step - loss: 0.6459 - accuracy: 0.6445 - val_loss: 0.6130 - val_accuracy: 0.6600

Epoch 7/40

63/63 [=====] - 39s 613ms/step - loss: 0.6414 - accuracy: 0.6510 - val_loss: 0.6497 - val_accuracy: 0.6270

Epoch 8/40

63/63 [=====] - 37s 595ms/step - loss: 0.6134 - accuracy: 0.6615 - val_loss: 0.6506 - val_accuracy: 0.6080

Epoch 9/40

63/63 [=====] - 41s 648ms/step - loss: 0.6196 - accuracy: 0.6630 - val_loss: 0.6355 - val_accuracy: 0.6260

Epoch 10/40

63/63 [=====] - 51s 818ms/step - loss: 0.6081 - accuracy: 0.6765 - val_loss: 0.6119 - val_accuracy: 0.6560

Epoch 11/40

63/63 [=====] - 51s 820ms/step - loss: 0.5998 - accuracy: 0.6770 - val_loss: 0.6042 - val_accuracy: 0.6710
Epoch 12/40
63/63 [=====] - 52s 820ms/step - loss: 0.5929 - accuracy: 0.6910 - val_loss: 0.5879 - val_accuracy: 0.6910
Epoch 13/40
63/63 [=====] - 52s 824ms/step - loss: 0.5999 - accuracy: 0.6830 - val_loss: 0.5726 - val_accuracy: 0.7120
Epoch 14/40
63/63 [=====] - 51s 809ms/step - loss: 0.5922 - accuracy: 0.6910 - val_loss: 0.5672 - val_accuracy: 0.7200
Epoch 15/40
63/63 [=====] - 51s 820ms/step - loss: 0.5857 - accuracy: 0.6980 - val_loss: 0.5796 - val_accuracy: 0.6970
Epoch 16/40
63/63 [=====] - 51s 807ms/step - loss: 0.5620 - accuracy: 0.6985 - val_loss: 0.5991 - val_accuracy: 0.7030
Epoch 17/40
63/63 [=====] - 51s 814ms/step - loss: 0.5682 - accuracy: 0.7185 - val_loss: 0.5584 - val_accuracy: 0.7430
Epoch 18/40
63/63 [=====] - 52s 825ms/step - loss: 0.5555 - accuracy: 0.7210 - val_loss: 0.5751 - val_accuracy: 0.6930
Epoch 19/40
63/63 [=====] - 51s 817ms/step - loss: 0.5469 - accuracy: 0.7290 - val_loss: 0.5809 - val_accuracy: 0.6770
Epoch 20/40
63/63 [=====] - 51s 818ms/step - loss: 0.5534 - accuracy: 0.7235 - val_loss: 0.5720 - val_accuracy: 0.7070
Epoch 21/40
63/63 [=====] - 51s 811ms/step - loss: 0.5313 - accuracy: 0.7370 - val_loss: 0.5750 - val_accuracy: 0.7410
Epoch 22/40
63/63 [=====] - 51s 814ms/step - loss: 0.5237 - accuracy: 0.7395 - val_loss: 0.5573 - val_accuracy: 0.7220
Epoch 23/40
63/63 [=====] - 51s 814ms/step - loss: 0.5298 - accuracy: 0.7320 - val_loss: 0.5565 - val_accuracy: 0.7260
Epoch 24/40
63/63 [=====] - 51s 811ms/step - loss: 0.5238 - accuracy: 0.7410 - val_loss: 0.4918 - val_accuracy: 0.7690
Epoch 25/40
63/63 [=====] - 52s 823ms/step - loss: 0.5055 - accuracy: 0.7625 - val_loss: 0.5557 - val_accuracy: 0.7250
Epoch 26/40
63/63 [=====] - 51s 814ms/step - loss: 0.5111 - accuracy: 0.7470 - val_loss: 0.5247 - val_accuracy: 0.7590
Epoch 27/40

```

63/63 [=====] - 52s 828ms/step - loss: 0.5066 -
accuracy: 0.7590 - val_loss: 0.5505 - val_accuracy: 0.7310
Epoch 28/40
63/63 [=====] - 51s 820ms/step - loss: 0.4809 -
accuracy: 0.7600 - val_loss: 0.4768 - val_accuracy: 0.7650
Epoch 29/40
63/63 [=====] - 51s 808ms/step - loss: 0.4872 -
accuracy: 0.7640 - val_loss: 0.4454 - val_accuracy: 0.7840
Epoch 30/40
63/63 [=====] - 51s 807ms/step - loss: 0.4824 -
accuracy: 0.7765 - val_loss: 0.4891 - val_accuracy: 0.7790
Epoch 31/40
63/63 [=====] - 51s 814ms/step - loss: 0.4571 -
accuracy: 0.7910 - val_loss: 0.6859 - val_accuracy: 0.7630
Epoch 32/40
63/63 [=====] - 51s 815ms/step - loss: 0.4875 -
accuracy: 0.7665 - val_loss: 0.4570 - val_accuracy: 0.7860
Epoch 33/40
63/63 [=====] - 51s 817ms/step - loss: 0.4782 -
accuracy: 0.7720 - val_loss: 0.5345 - val_accuracy: 0.7450
Epoch 34/40
63/63 [=====] - 51s 815ms/step - loss: 0.4632 -
accuracy: 0.7890 - val_loss: 0.5078 - val_accuracy: 0.7620
Epoch 35/40
63/63 [=====] - 51s 811ms/step - loss: 0.4491 -
accuracy: 0.7960 - val_loss: 0.5608 - val_accuracy: 0.7380
Epoch 36/40
63/63 [=====] - 51s 811ms/step - loss: 0.4545 -
accuracy: 0.7895 - val_loss: 0.5096 - val_accuracy: 0.7850
Epoch 37/40
63/63 [=====] - 51s 813ms/step - loss: 0.4572 -
accuracy: 0.8000 - val_loss: 0.5641 - val_accuracy: 0.7350
Epoch 38/40
63/63 [=====] - 51s 818ms/step - loss: 0.4517 -
accuracy: 0.7865 - val_loss: 0.4791 - val_accuracy: 0.7940
Epoch 39/40
63/63 [=====] - 54s 857ms/step - loss: 0.4378 -
accuracy: 0.8070 - val_loss: 0.4361 - val_accuracy: 0.7950
Epoch 40/40
63/63 [=====] - 50s 795ms/step - loss: 0.4639 -
accuracy: 0.7865 - val_loss: 0.5188 - val_accuracy: 0.7900

```

```

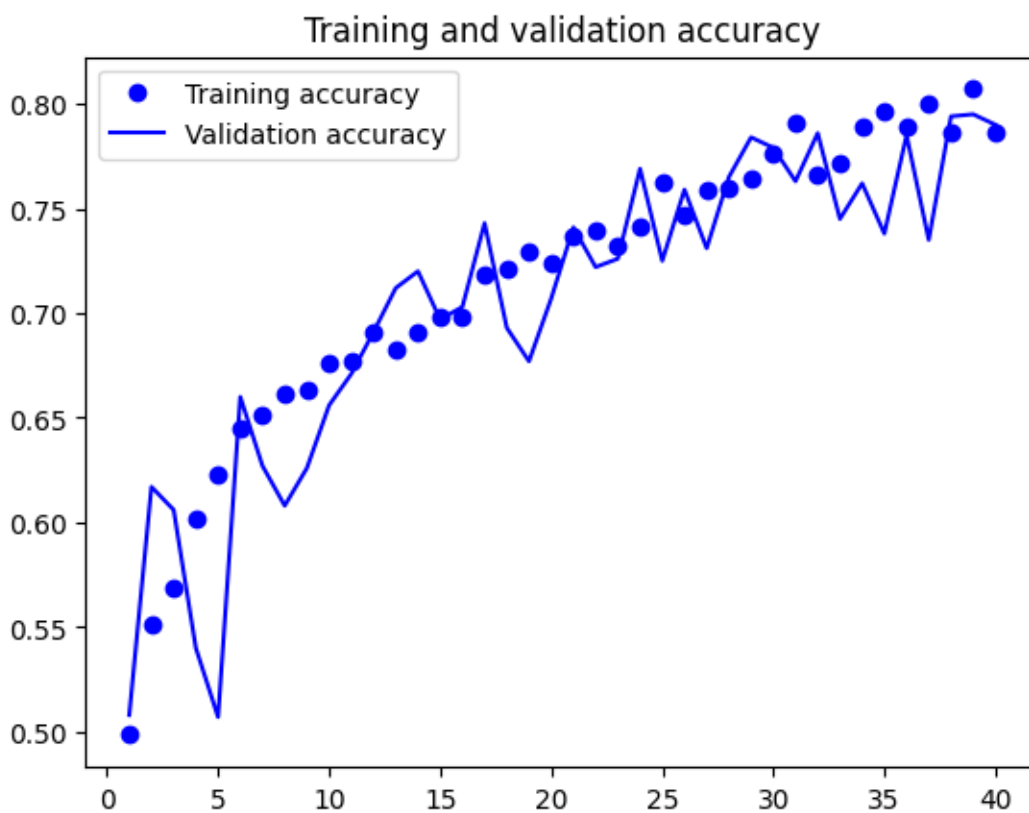
[82]: # plot the performance
import matplotlib.pyplot as plt
accuracy = history.history["accuracy"]
val_accuracy = history.history["val_accuracy"]
loss = history.history["loss"]

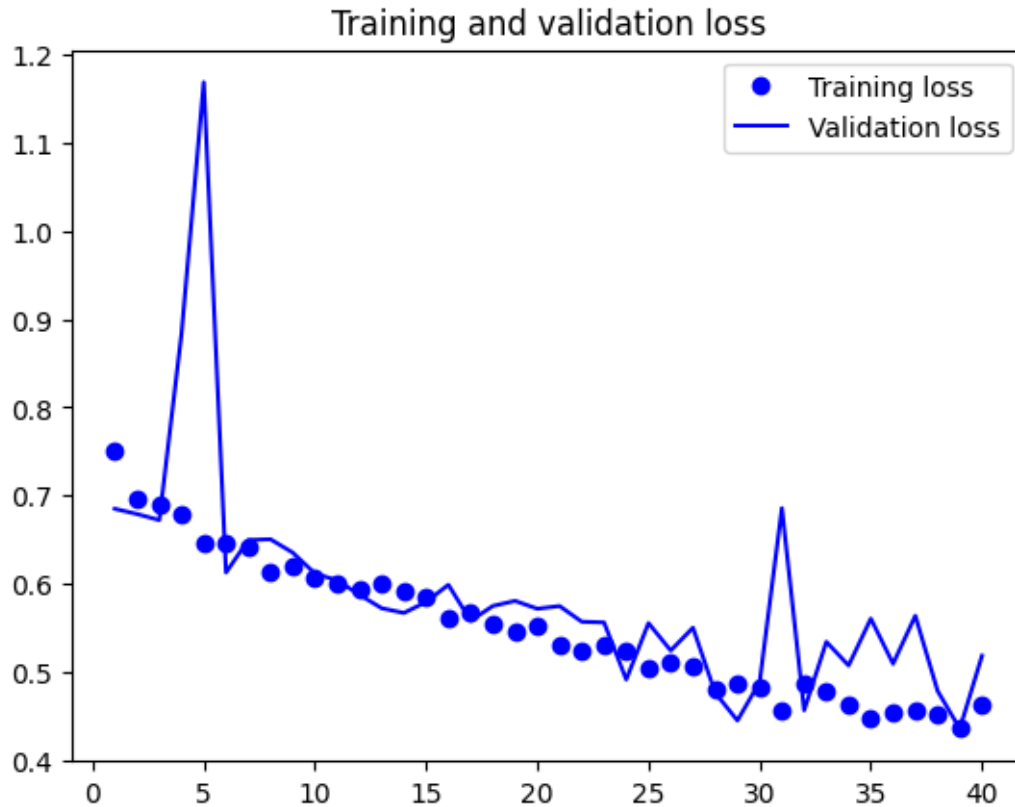
```

```

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

```





```
[83]: # Evaluating the model on the test set
```

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

```
# scratch model done by here
```

```
32/32 [=====] - 4s 106ms/step - loss: 0.5208 -
accuracy: 0.7720
Test accuracy: 0.772
```

```
[ ]: # pretrain model
```

```
[ ]: # part1 train/vali/test 1000 500 500
```

```
[84]: # Instantiating the VGG16 convolutional base
```

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

```
input_shape=(180, 180, 3))
conv_base.summary()
```

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_10 (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: 14,714,688		
Trainable params: 14,714,688		

Non-trainable params: 0

```
[85]: # Fast feature extraction without data augmentation
import numpy as np

def get_features_and_labels(dataset):
    all_features = []
    all_labels = []
    for images, labels in dataset:
        preprocessed_images = keras.applications.vgg16.preprocess_input(images)
        features = conv_base.predict(preprocessed_images)
        all_features.append(features)
        all_labels.append(labels)
    return np.concatenate(all_features), np.concatenate(all_labels)

train_features, train_labels = get_features_and_labels(train_dataset)
val_features, val_labels = get_features_and_labels(validation_dataset)
test_features, test_labels = get_features_and_labels(test_dataset)
train_features.shape
# The extracted features are currently of shape (samples, 5, 5, 512):
```

```
1/1 [=====] - 10s 10s/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 13ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 26ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 13ms/step
1/1 [=====] - 0s 26ms/step
1/1 [=====] - 0s 13ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 19ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 13ms/step
1/1 [=====] - 0s 12ms/step
```

```

1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 13ms/step
1/1 [=====] - 0s 13ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 19ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 20ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 19ms/step
1/1 [=====] - 0s 19ms/step
1/1 [=====] - 0s 19ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 11ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 33ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 19ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 19ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 8s 8s/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 16ms/step

```


1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 13ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 12ms/step
1/1 [=====] - 0s 26ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 8ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 26ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 6s 6s/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 13ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 9ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 12ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 16ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 14ms/step

```

1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 12ms/step
1/1 [=====] - 0s 15ms/step
1/1 [=====] - 0s 29ms/step
1/1 [=====] - 0s 15ms/step

```

[85]: (2000, 5, 5, 512)

[86]: *# Defining and training the densely connected classifier*

```

inputs = keras.Input(shape=(5, 5, 512))
x = layers.Flatten()(inputs)
x = layers.Dense(256)(x)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs, outputs)
model.compile(loss="binary_crossentropy",
              optimizer="rmsprop",
              metrics=["accuracy"])

callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="feature_extraction.keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_features, train_labels,
    epochs=20,
    validation_data=(val_features, val_labels),
    callbacks=callbacks)

```

Epoch 1/20

```

63/63 [=====] - 3s 40ms/step - loss: 13.7310 -
accuracy: 0.9270 - val_loss: 6.2004 - val_accuracy: 0.9510

```

Epoch 2/20

```

63/63 [=====] - 2s 38ms/step - loss: 3.8567 - accuracy:
0.9715 - val_loss: 7.7510 - val_accuracy: 0.9600

```

Epoch 3/20

```

63/63 [=====] - 2s 38ms/step - loss: 1.9190 - accuracy:
0.9860 - val_loss: 4.4973 - val_accuracy: 0.9690

```

Epoch 4/20

```

63/63 [=====] - 2s 37ms/step - loss: 0.8778 - accuracy:
0.9920 - val_loss: 7.8846 - val_accuracy: 0.9630

```

Epoch 5/20

```

63/63 [=====] - 2s 37ms/step - loss: 1.1901 - accuracy:
0.9910 - val_loss: 4.8347 - val_accuracy: 0.9710

```

Epoch 6/20

```

63/63 [=====] - 2s 38ms/step - loss: 0.5795 - accuracy:

```

```

0.9960 - val_loss: 6.2593 - val_accuracy: 0.9740
Epoch 7/20
63/63 [=====] - 2s 38ms/step - loss: 0.3022 - accuracy:
0.9965 - val_loss: 5.0825 - val_accuracy: 0.9760
Epoch 8/20
63/63 [=====] - 2s 37ms/step - loss: 1.0407 - accuracy:
0.9900 - val_loss: 9.4600 - val_accuracy: 0.9670
Epoch 9/20
63/63 [=====] - 2s 37ms/step - loss: 0.2900 - accuracy:
0.9960 - val_loss: 5.2798 - val_accuracy: 0.9700
Epoch 10/20
63/63 [=====] - 2s 37ms/step - loss: 0.6306 - accuracy:
0.9950 - val_loss: 5.9452 - val_accuracy: 0.9720
Epoch 11/20
63/63 [=====] - 2s 37ms/step - loss: 0.6491 - accuracy:
0.9945 - val_loss: 7.2021 - val_accuracy: 0.9710
Epoch 12/20
63/63 [=====] - 2s 37ms/step - loss: 0.1005 - accuracy:
0.9995 - val_loss: 5.4700 - val_accuracy: 0.9770
Epoch 13/20
63/63 [=====] - 2s 38ms/step - loss: 0.1096 - accuracy:
0.9975 - val_loss: 6.0828 - val_accuracy: 0.9780
Epoch 14/20
63/63 [=====] - 2s 38ms/step - loss: 0.2622 - accuracy:
0.9980 - val_loss: 5.8576 - val_accuracy: 0.9790
Epoch 15/20
63/63 [=====] - 2s 37ms/step - loss: 0.3693 - accuracy:
0.9985 - val_loss: 7.6751 - val_accuracy: 0.9720
Epoch 16/20
63/63 [=====] - 2s 37ms/step - loss: 0.0384 - accuracy:
0.9995 - val_loss: 5.2329 - val_accuracy: 0.9780
Epoch 17/20
63/63 [=====] - 2s 39ms/step - loss: 1.2258e-17 -
accuracy: 1.0000 - val_loss: 5.2329 - val_accuracy: 0.9780
Epoch 18/20
63/63 [=====] - 2s 38ms/step - loss: 3.9445e-23 -
accuracy: 1.0000 - val_loss: 5.2329 - val_accuracy: 0.9780
Epoch 19/20
63/63 [=====] - 2s 37ms/step - loss: 3.3851e-29 -
accuracy: 1.0000 - val_loss: 5.2329 - val_accuracy: 0.9780
Epoch 20/20
63/63 [=====] - 2s 37ms/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 5.2329 - val_accuracy: 0.9780

```

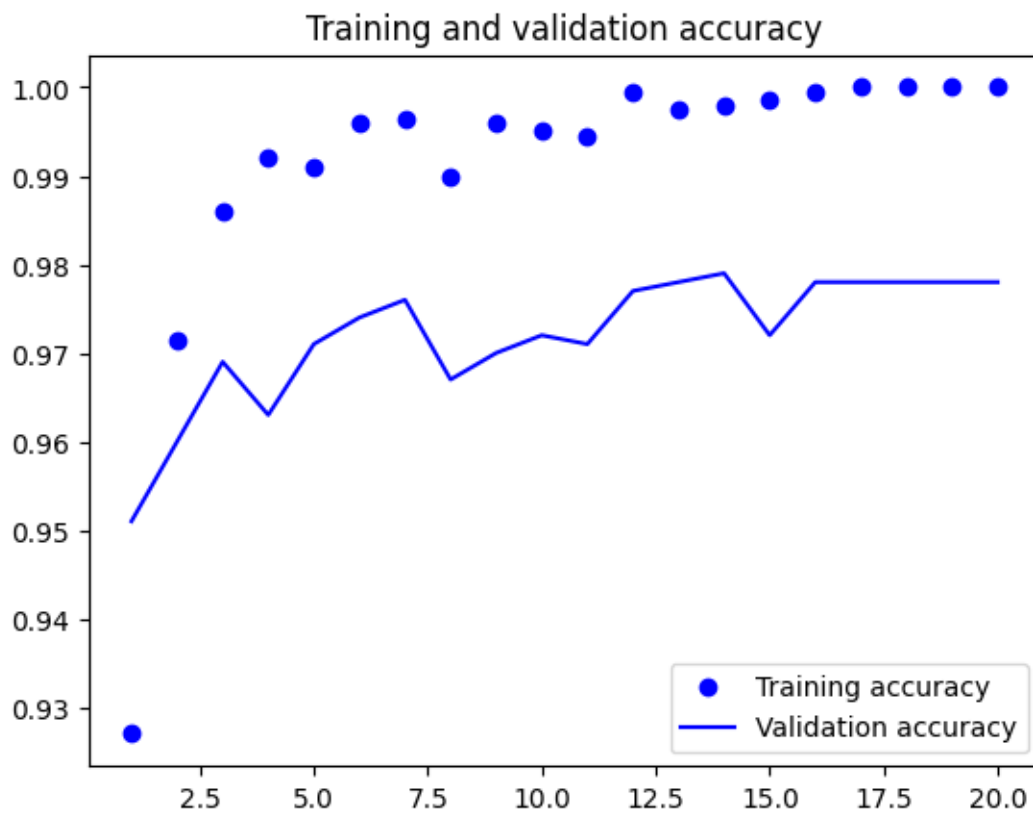
```
[87]: # Plotting the results
```

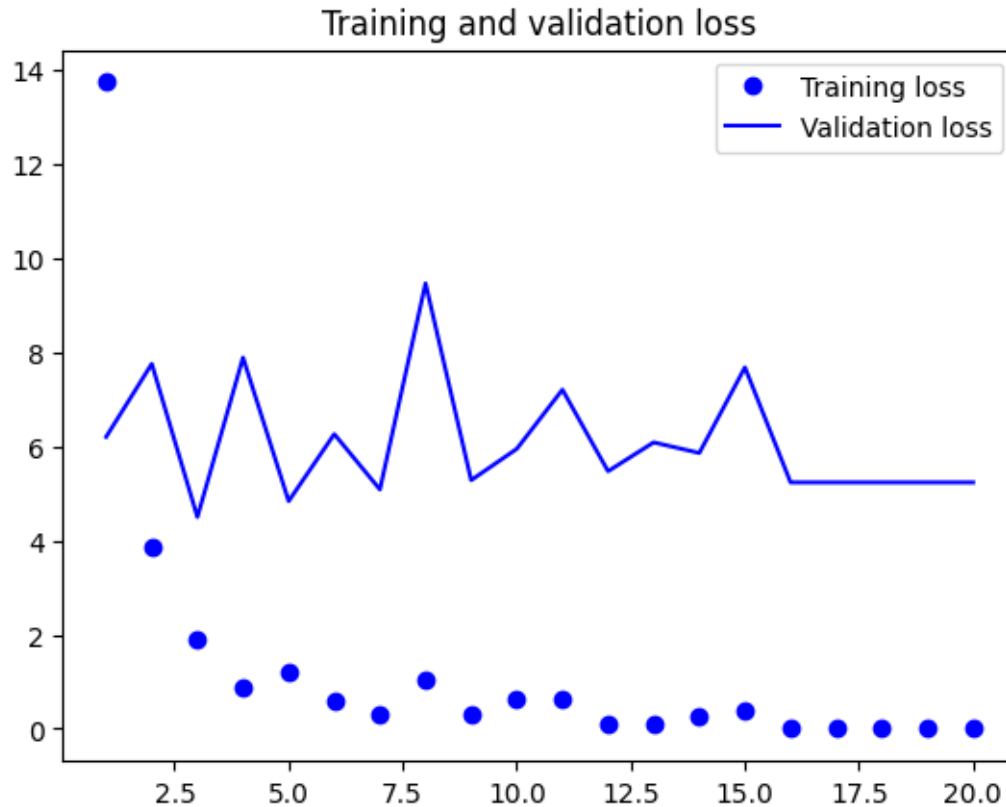
```
import matplotlib.pyplot as plt
```

```

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

```





```
[88]: test_model = keras.models.load_model(
        "convnet_from_scratch_with_augmentation.keras")
test_loss, test_acc = test_model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")
```

```
32/32 [=====] - 4s 101ms/step - loss: 0.5208 -
accuracy: 0.7720
Test accuracy: 0.772
```

```
[89]: # Feature extraction together with data augmentation
# Instantiating and freezing the VGG16 convolutional base

conv_base = keras.applications.vgg16.VGG16(
    weights="imagenet",
    include_top=False)
conv_base.trainable = False

# Printing the list of trainable weights before and after freezing

conv_base.trainable = True
print("This is the number of trainable weights ")
```

```

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

[90]: *# Adding a data augmentation stage and a classifier to the convolutional base*

```

data_augmentation = keras.Sequential(
    [
        layers.RandomFlip("horizontal"),
        layers.RandomRotation(0.1),
        layers.RandomZoom(0.2),
    ]
)

inputs = keras.Input(shape=(180, 180, 3))
x = data_augmentation(inputs)
x = keras.applications.vgg16.preprocess_input(x)
x = conv_base(x)
x = layers.Flatten()(x)
x = layers.Dense(256)(x)
x = layers.Dropout(0.5)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs, outputs)
model.compile(loss="binary_crossentropy",
              optimizer="rmsprop",
              metrics=["accuracy"])

callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="feature_extraction_with_data_augmentation.keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=50,
    validation_data=validation_dataset,
    callbacks=callbacks)

# Evaluating the model on the test set

test_model = keras.models.load_model(
    "feature_extraction_with_data_augmentation.keras")

```

```
test_loss, test_acc = test_model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")
# performance on test is great!!
```

Epoch 1/50

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

63/63 [=====] - 26s 378ms/step - loss: 19.5066 -

accuracy: 0.8940 - val_loss: 4.7592 - val_accuracy: 0.9660

Epoch 2/50

63/63 [=====] - 23s 366ms/step - loss: 7.3505 -
accuracy: 0.9480 - val_loss: 4.0718 - val_accuracy: 0.9730
Epoch 3/50
63/63 [=====] - 23s 373ms/step - loss: 6.6962 -
accuracy: 0.9560 - val_loss: 4.7772 - val_accuracy: 0.9670
Epoch 4/50
63/63 [=====] - 23s 365ms/step - loss: 4.5027 -
accuracy: 0.9640 - val_loss: 2.4487 - val_accuracy: 0.9840
Epoch 5/50
63/63 [=====] - 24s 377ms/step - loss: 3.3729 -
accuracy: 0.9765 - val_loss: 6.0196 - val_accuracy: 0.9730
Epoch 6/50
63/63 [=====] - 23s 370ms/step - loss: 3.9285 -
accuracy: 0.9700 - val_loss: 11.8763 - val_accuracy: 0.9560
Epoch 7/50
63/63 [=====] - 23s 370ms/step - loss: 2.8783 -
accuracy: 0.9740 - val_loss: 4.7570 - val_accuracy: 0.9810
Epoch 8/50
63/63 [=====] - 23s 361ms/step - loss: 1.6970 -
accuracy: 0.9800 - val_loss: 5.4612 - val_accuracy: 0.9780
Epoch 9/50
63/63 [=====] - 23s 367ms/step - loss: 2.4427 -
accuracy: 0.9765 - val_loss: 3.3062 - val_accuracy: 0.9810
Epoch 10/50
63/63 [=====] - 23s 364ms/step - loss: 1.7015 -
accuracy: 0.9780 - val_loss: 5.4824 - val_accuracy: 0.9720
Epoch 11/50
63/63 [=====] - 24s 375ms/step - loss: 1.4910 -
accuracy: 0.9845 - val_loss: 4.0360 - val_accuracy: 0.9760
Epoch 12/50
63/63 [=====] - 23s 370ms/step - loss: 1.5672 -
accuracy: 0.9865 - val_loss: 4.7089 - val_accuracy: 0.9780
Epoch 13/50
63/63 [=====] - 23s 371ms/step - loss: 1.8037 -
accuracy: 0.9825 - val_loss: 3.1303 - val_accuracy: 0.9760
Epoch 14/50
63/63 [=====] - 23s 370ms/step - loss: 1.9998 -
accuracy: 0.9795 - val_loss: 4.4065 - val_accuracy: 0.9770
Epoch 15/50
63/63 [=====] - 23s 372ms/step - loss: 1.4778 -
accuracy: 0.9845 - val_loss: 4.5979 - val_accuracy: 0.9790
Epoch 16/50
63/63 [=====] - 24s 379ms/step - loss: 1.6827 -
accuracy: 0.9820 - val_loss: 4.2264 - val_accuracy: 0.9710
Epoch 17/50
63/63 [=====] - 25s 391ms/step - loss: 0.9366 -
accuracy: 0.9870 - val_loss: 3.4441 - val_accuracy: 0.9810
Epoch 18/50

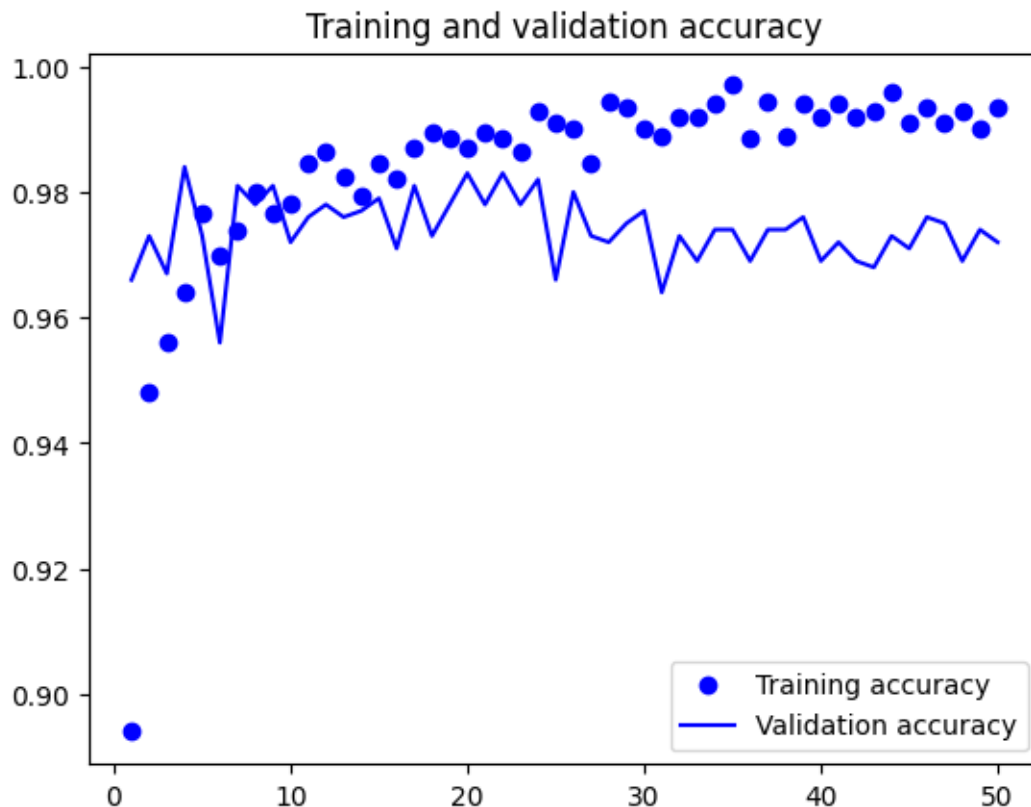
63/63 [=====] - 24s 383ms/step - loss: 0.8752 - accuracy: 0.9895 - val_loss: 3.7345 - val_accuracy: 0.9730
Epoch 19/50
63/63 [=====] - 25s 390ms/step - loss: 0.9341 - accuracy: 0.9885 - val_loss: 3.9705 - val_accuracy: 0.9780
Epoch 20/50
63/63 [=====] - 24s 382ms/step - loss: 1.2534 - accuracy: 0.9870 - val_loss: 3.6498 - val_accuracy: 0.9830
Epoch 21/50
63/63 [=====] - 25s 394ms/step - loss: 1.2249 - accuracy: 0.9895 - val_loss: 3.6458 - val_accuracy: 0.9780
Epoch 22/50
63/63 [=====] - 24s 382ms/step - loss: 0.9485 - accuracy: 0.9885 - val_loss: 3.0947 - val_accuracy: 0.9830
Epoch 23/50
63/63 [=====] - 24s 389ms/step - loss: 1.2505 - accuracy: 0.9865 - val_loss: 3.0575 - val_accuracy: 0.9780
Epoch 24/50
63/63 [=====] - 24s 386ms/step - loss: 0.7461 - accuracy: 0.9930 - val_loss: 2.9054 - val_accuracy: 0.9820
Epoch 25/50
63/63 [=====] - 25s 399ms/step - loss: 0.8112 - accuracy: 0.9910 - val_loss: 6.1922 - val_accuracy: 0.9660
Epoch 26/50
63/63 [=====] - 25s 390ms/step - loss: 0.9524 - accuracy: 0.9900 - val_loss: 3.4678 - val_accuracy: 0.9800
Epoch 27/50
63/63 [=====] - 25s 399ms/step - loss: 1.2549 - accuracy: 0.9845 - val_loss: 4.9630 - val_accuracy: 0.9730
Epoch 28/50
63/63 [=====] - 25s 390ms/step - loss: 0.2379 - accuracy: 0.9945 - val_loss: 3.4087 - val_accuracy: 0.9720
Epoch 29/50
63/63 [=====] - 25s 394ms/step - loss: 0.3468 - accuracy: 0.9935 - val_loss: 4.2364 - val_accuracy: 0.9750
Epoch 30/50
63/63 [=====] - 24s 386ms/step - loss: 0.6393 - accuracy: 0.9900 - val_loss: 2.8140 - val_accuracy: 0.9770
Epoch 31/50
63/63 [=====] - 25s 392ms/step - loss: 0.7802 - accuracy: 0.9890 - val_loss: 7.4862 - val_accuracy: 0.9640
Epoch 32/50
63/63 [=====] - 24s 382ms/step - loss: 0.5153 - accuracy: 0.9920 - val_loss: 5.4123 - val_accuracy: 0.9730
Epoch 33/50
63/63 [=====] - 24s 389ms/step - loss: 0.8305 - accuracy: 0.9920 - val_loss: 4.7212 - val_accuracy: 0.9690
Epoch 34/50

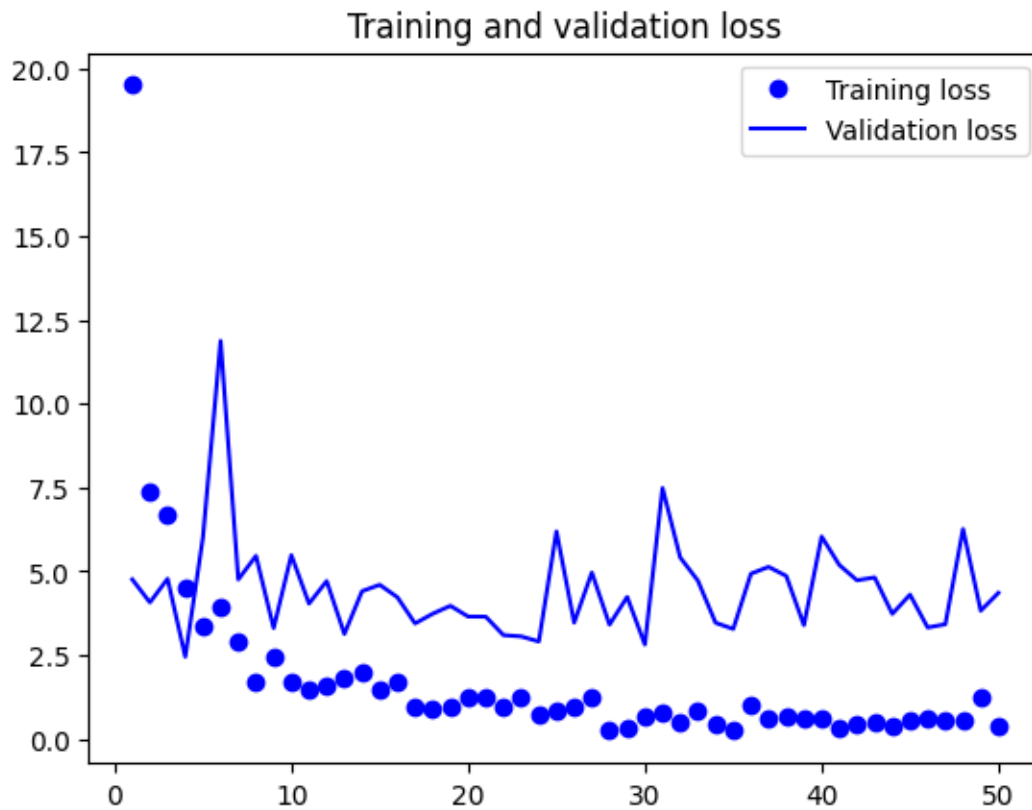
63/63 [=====] - 24s 385ms/step - loss: 0.4498 - accuracy: 0.9940 - val_loss: 3.4641 - val_accuracy: 0.9740
Epoch 35/50
63/63 [=====] - 25s 394ms/step - loss: 0.2787 - accuracy: 0.9970 - val_loss: 3.2806 - val_accuracy: 0.9740
Epoch 36/50
63/63 [=====] - 24s 386ms/step - loss: 1.0130 - accuracy: 0.9885 - val_loss: 4.9279 - val_accuracy: 0.9690
Epoch 37/50
63/63 [=====] - 24s 389ms/step - loss: 0.6028 - accuracy: 0.9945 - val_loss: 5.1325 - val_accuracy: 0.9740
Epoch 38/50
63/63 [=====] - 24s 382ms/step - loss: 0.6761 - accuracy: 0.9890 - val_loss: 4.8611 - val_accuracy: 0.9740
Epoch 39/50
63/63 [=====] - 24s 389ms/step - loss: 0.6149 - accuracy: 0.9940 - val_loss: 3.3927 - val_accuracy: 0.9760
Epoch 40/50
63/63 [=====] - 24s 389ms/step - loss: 0.6326 - accuracy: 0.9920 - val_loss: 6.0380 - val_accuracy: 0.9690
Epoch 41/50
63/63 [=====] - 25s 397ms/step - loss: 0.3334 - accuracy: 0.9940 - val_loss: 5.1891 - val_accuracy: 0.9720
Epoch 42/50
63/63 [=====] - 25s 393ms/step - loss: 0.4573 - accuracy: 0.9920 - val_loss: 4.7317 - val_accuracy: 0.9690
Epoch 43/50
63/63 [=====] - 25s 404ms/step - loss: 0.5137 - accuracy: 0.9930 - val_loss: 4.8168 - val_accuracy: 0.9680
Epoch 44/50
63/63 [=====] - 24s 387ms/step - loss: 0.3659 - accuracy: 0.9960 - val_loss: 3.7299 - val_accuracy: 0.9730
Epoch 45/50
63/63 [=====] - 25s 395ms/step - loss: 0.5773 - accuracy: 0.9910 - val_loss: 4.3044 - val_accuracy: 0.9710
Epoch 46/50
63/63 [=====] - 24s 388ms/step - loss: 0.5943 - accuracy: 0.9935 - val_loss: 3.3227 - val_accuracy: 0.9760
Epoch 47/50
63/63 [=====] - 25s 391ms/step - loss: 0.5700 - accuracy: 0.9910 - val_loss: 3.4212 - val_accuracy: 0.9750
Epoch 48/50
63/63 [=====] - 24s 384ms/step - loss: 0.5803 - accuracy: 0.9930 - val_loss: 6.2636 - val_accuracy: 0.9690
Epoch 49/50
63/63 [=====] - 24s 386ms/step - loss: 1.2185 - accuracy: 0.9900 - val_loss: 3.8250 - val_accuracy: 0.9740
Epoch 50/50

```
63/63 [=====] - 24s 381ms/step - loss: 0.3873 -  
accuracy: 0.9935 - val_loss: 4.3566 - val_accuracy: 0.9720  
32/32 [=====] - 1s 41ms/step - loss: 5.3369 - accuracy:  
0.9730  
Test accuracy: 0.973
```

```
[91]: # Plotting the results
```

```
import matplotlib.pyplot as plt  
acc = history.history["accuracy"]  
val_acc = history.history["val_accuracy"]  
loss = history.history["loss"]  
val_loss = history.history["val_loss"]  
epochs = range(1, len(acc) + 1)  
plt.plot(epochs, acc, "bo", label="Training accuracy")  
plt.plot(epochs, val_acc, "b", label="Validation accuracy")  
plt.title("Training and validation accuracy")  
plt.legend()  
plt.figure()  
plt.plot(epochs, loss, "bo", label="Training loss")  
plt.plot(epochs, val_loss, "b", label="Validation loss")  
plt.title("Training and validation loss")  
plt.legend()  
plt.show()
```





```
[92]: # Fine-tuning a pretrained model

conv_base.summary()

# Freezing all layers until the fourth from the last

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

# Fine-tuning the model

model.compile(loss="binary_crossentropy",
              optimizer=keras.optimizers.RMSprop(learning_rate=1e-5),
              metrics=["accuracy"])

callbacks = [
    keras.callbacks.ModelCheckpoint(
```

```

        filepath="fine_tuning.keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=30,
    validation_data=validation_dataset,
    callbacks=callbacks)
model = keras.models.load_model("fine_tuning.keras")
test_loss, test_acc = model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")

# Plotting the results

import matplotlib.pyplot as plt
acc = history.history["accuracy"]
val_acc = history.history["val_accuracy"]
loss = history.history["loss"]
val_loss = history.history["val_loss"]
epochs = range(1, len(acc) + 1)
plt.plot(epochs, acc, "bo", label="Training accuracy")
plt.plot(epochs, val_acc, "b", label="Validation accuracy")
plt.title("Training and validation accuracy")
plt.legend()
plt.figure()
plt.plot(epochs, loss, "bo", label="Training loss")
plt.plot(epochs, val_loss, "b", label="Validation loss")
plt.title("Training and validation loss")
plt.legend()
plt.show()

```

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_12 (InputLayer)	[(None, None, None, 3)]	0
block1_conv1 (Conv2D)	(None, None, None, 64)	1792
block1_conv2 (Conv2D)	(None, None, None, 64)	36928
block1_pool (MaxPooling2D)	(None, None, None, 64)	0
block2_conv1 (Conv2D)	(None, None, None, 128)	73856
block2_conv2 (Conv2D)	(None, None, None, 128)	147584

block2_pool (MaxPooling2D)	(None, None, None, 128)	0
block3_conv1 (Conv2D)	(None, None, None, 256)	295168
block3_conv2 (Conv2D)	(None, None, None, 256)	590080
block3_conv3 (Conv2D)	(None, None, None, 256)	590080
block3_pool (MaxPooling2D)	(None, None, None, 256)	0
block4_conv1 (Conv2D)	(None, None, None, 512)	1180160
block4_conv2 (Conv2D)	(None, None, None, 512)	2359808
block4_conv3 (Conv2D)	(None, None, None, 512)	2359808
block4_pool (MaxPooling2D)	(None, None, None, 512)	0
block5_conv1 (Conv2D)	(None, None, None, 512)	2359808
block5_conv2 (Conv2D)	(None, None, None, 512)	2359808
block5_conv3 (Conv2D)	(None, None, None, 512)	2359808
block5_pool (MaxPooling2D)	(None, None, None, 512)	0

```

=====
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688

```

```

-----
Epoch 1/30

```

```

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3
cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there
is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no
registered converter for this op.

```

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

63/63 [=====] - 22s 305ms/step - loss: 3.1957 - accuracy: 0.9710 - val_loss: 5.1517 - val_accuracy: 0.9690

Epoch 2/30

63/63 [=====] - 18s 291ms/step - loss: 1.7003 - accuracy: 0.9780 - val_loss: 3.1216 - val_accuracy: 0.9790

Epoch 3/30

63/63 [=====] - 18s 291ms/step - loss: 1.1426 - accuracy: 0.9825 - val_loss: 3.8116 - val_accuracy: 0.9740

Epoch 4/30

63/63 [=====] - 18s 287ms/step - loss: 1.3489 - accuracy: 0.9805 - val_loss: 2.7680 - val_accuracy: 0.9760

Epoch 5/30

63/63 [=====] - 18s 290ms/step - loss: 0.8155 - accuracy: 0.9820 - val_loss: 2.5178 - val_accuracy: 0.9770

Epoch 6/30

63/63 [=====] - 18s 286ms/step - loss: 0.6679 - accuracy: 0.9895 - val_loss: 3.1287 - val_accuracy: 0.9710

Epoch 7/30

63/63 [=====] - 18s 287ms/step - loss: 0.6936 - accuracy: 0.9875 - val_loss: 2.8682 - val_accuracy: 0.9730

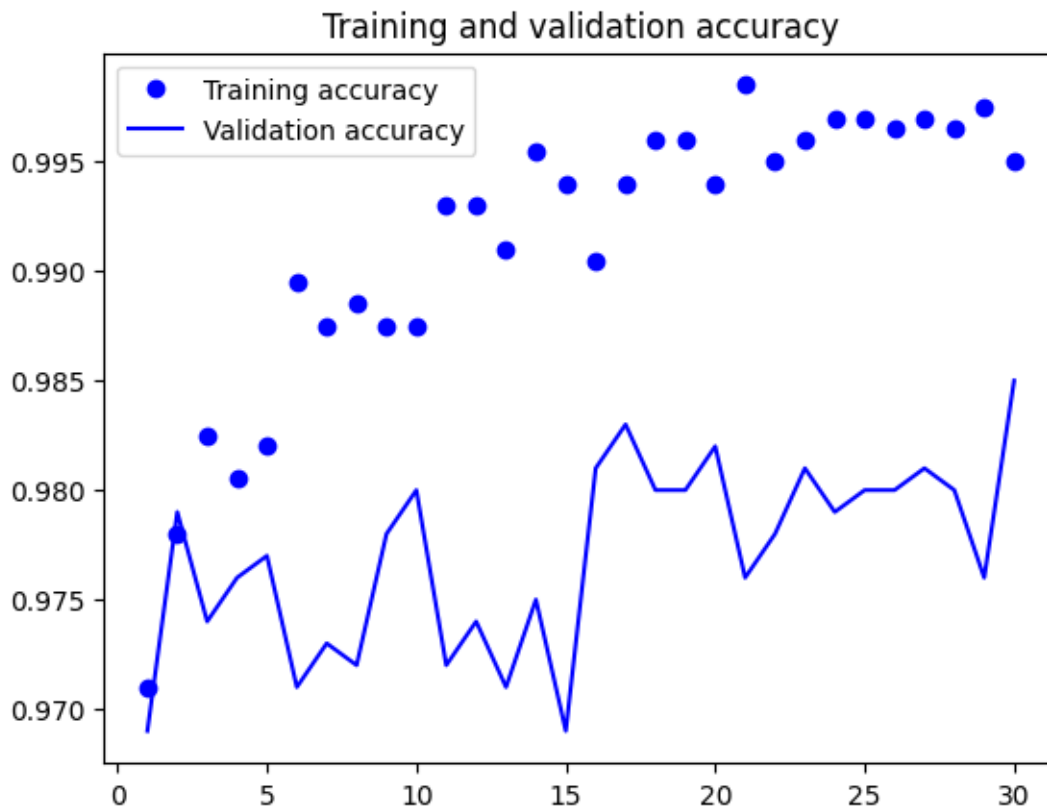
Epoch 8/30

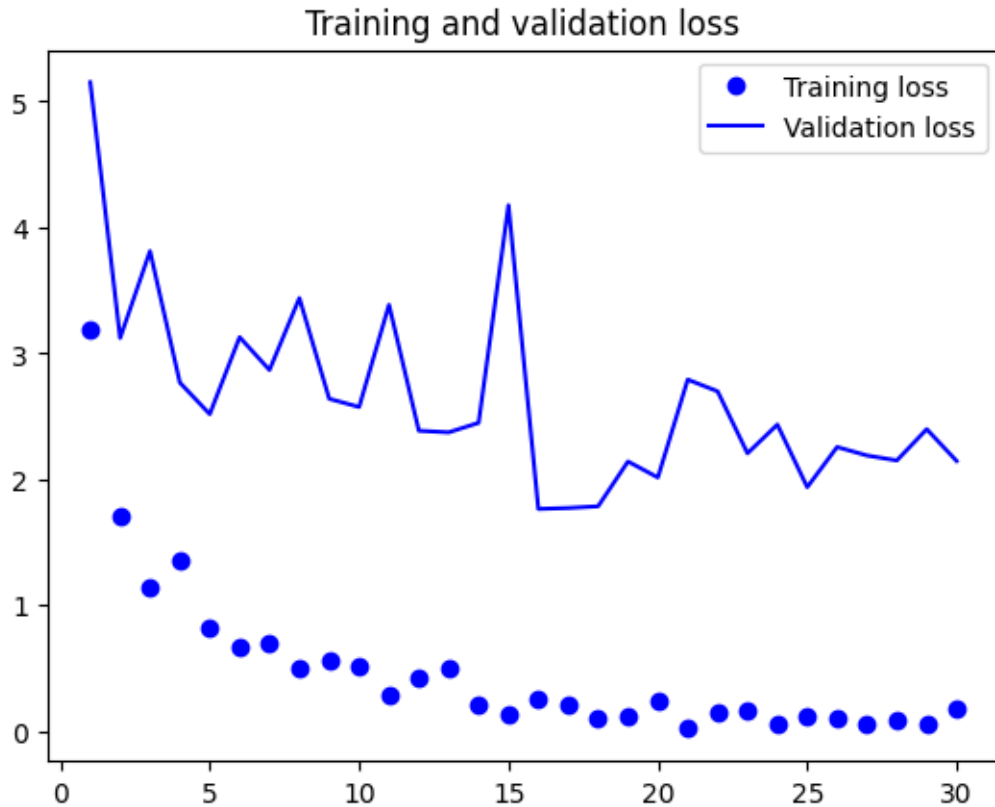
63/63 [=====] - 18s 289ms/step - loss: 0.4988 - accuracy: 0.9885 - val_loss: 3.4389 - val_accuracy: 0.9720

Epoch 9/30

63/63 [=====] - 19s 295ms/step - loss: 0.5674 -
accuracy: 0.9875 - val_loss: 2.6394 - val_accuracy: 0.9780
Epoch 10/30
63/63 [=====] - 19s 294ms/step - loss: 0.5176 -
accuracy: 0.9875 - val_loss: 2.5744 - val_accuracy: 0.9800
Epoch 11/30
63/63 [=====] - 19s 295ms/step - loss: 0.2868 -
accuracy: 0.9930 - val_loss: 3.3852 - val_accuracy: 0.9720
Epoch 12/30
63/63 [=====] - 18s 293ms/step - loss: 0.4319 -
accuracy: 0.9930 - val_loss: 2.3860 - val_accuracy: 0.9740
Epoch 13/30
63/63 [=====] - 18s 292ms/step - loss: 0.4980 -
accuracy: 0.9910 - val_loss: 2.3732 - val_accuracy: 0.9710
Epoch 14/30
63/63 [=====] - 18s 286ms/step - loss: 0.2054 -
accuracy: 0.9955 - val_loss: 2.4488 - val_accuracy: 0.9750
Epoch 15/30
63/63 [=====] - 19s 300ms/step - loss: 0.1364 -
accuracy: 0.9940 - val_loss: 4.1756 - val_accuracy: 0.9690
Epoch 16/30
63/63 [=====] - 19s 296ms/step - loss: 0.2537 -
accuracy: 0.9905 - val_loss: 1.7653 - val_accuracy: 0.9810
Epoch 17/30
63/63 [=====] - 19s 294ms/step - loss: 0.2165 -
accuracy: 0.9940 - val_loss: 1.7728 - val_accuracy: 0.9830
Epoch 18/30
63/63 [=====] - 18s 285ms/step - loss: 0.0969 -
accuracy: 0.9960 - val_loss: 1.7860 - val_accuracy: 0.9800
Epoch 19/30
63/63 [=====] - 18s 288ms/step - loss: 0.1141 -
accuracy: 0.9960 - val_loss: 2.1420 - val_accuracy: 0.9800
Epoch 20/30
63/63 [=====] - 18s 286ms/step - loss: 0.2463 -
accuracy: 0.9940 - val_loss: 2.0149 - val_accuracy: 0.9820
Epoch 21/30
63/63 [=====] - 18s 293ms/step - loss: 0.0205 -
accuracy: 0.9985 - val_loss: 2.7919 - val_accuracy: 0.9760
Epoch 22/30
63/63 [=====] - 18s 291ms/step - loss: 0.1542 -
accuracy: 0.9950 - val_loss: 2.6988 - val_accuracy: 0.9780
Epoch 23/30
63/63 [=====] - 18s 289ms/step - loss: 0.1643 -
accuracy: 0.9960 - val_loss: 2.2068 - val_accuracy: 0.9810
Epoch 24/30
63/63 [=====] - 18s 287ms/step - loss: 0.0532 -
accuracy: 0.9970 - val_loss: 2.4348 - val_accuracy: 0.9790
Epoch 25/30

63/63 [=====] - 18s 287ms/step - loss: 0.1215 - accuracy: 0.9970 - val_loss: 1.9383 - val_accuracy: 0.9800
Epoch 26/30
63/63 [=====] - 18s 287ms/step - loss: 0.0984 - accuracy: 0.9965 - val_loss: 2.2567 - val_accuracy: 0.9800
Epoch 27/30
63/63 [=====] - 18s 286ms/step - loss: 0.0612 - accuracy: 0.9970 - val_loss: 2.1894 - val_accuracy: 0.9810
Epoch 28/30
63/63 [=====] - 19s 295ms/step - loss: 0.0954 - accuracy: 0.9965 - val_loss: 2.1502 - val_accuracy: 0.9800
Epoch 29/30
63/63 [=====] - 19s 297ms/step - loss: 0.0647 - accuracy: 0.9975 - val_loss: 2.3985 - val_accuracy: 0.9760
Epoch 30/30
63/63 [=====] - 19s 295ms/step - loss: 0.1784 - accuracy: 0.9950 - val_loss: 2.1460 - val_accuracy: 0.9850
32/32 [=====] - 2s 53ms/step - loss: 2.6329 - accuracy: 0.9750
Test accuracy: 0.975





```
[ ]: # fit-tuning with train/val/test 3000 750 750
```

```
[93]: original_dir = pathlib.Path('C:\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\train')
new_base_dir = pathlib.Path('C:
↳\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\cats_vs_dogs_hw2_fit_model4')
```

```
def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = new_base_dir / subset_name / category
        os.makedirs(dir)
        fnames = [f"{category}.{i}.jpg"
                    for i in range(start_index, end_index)]
        for fname in fnames:
            shutil.copyfile(src=original_dir / fname,
                            dst=dir / fname)
make_subset("train", start_index=0, end_index=3000)          ###3000  ␣
↳###      3000 6000
make_subset("validation", start_index=3000, end_index=3750) ###750
make_subset("test", start_index=3750, end_index=4500)       ###750 TEST
```

```

[95]: # Fine-tuning a pretrained model

conv_base.summary()

# Freezing all layers until the fourth from the last

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

# Fine-tuning the model

model.compile(loss="binary_crossentropy",
              optimizer=keras.optimizers.RMSprop(learning_rate=1e-5),
              metrics=["accuracy"])

callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="fine_tuning.keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=30,
    validation_data=validation_dataset,
    callbacks=callbacks)
model = keras.models.load_model("fine_tuning.keras")
test_loss, test_acc = model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")

# Plotting the results

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

```

```
plt.legend()
plt.show()
```

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_12 (InputLayer)	[(None, None, None, 3)]	0
block1_conv1 (Conv2D)	(None, None, None, 64)	1792
block1_conv2 (Conv2D)	(None, None, None, 64)	36928
block1_pool (MaxPooling2D)	(None, None, None, 64)	0
block2_conv1 (Conv2D)	(None, None, None, 128)	73856
block2_conv2 (Conv2D)	(None, None, None, 128)	147584
block2_pool (MaxPooling2D)	(None, None, None, 128)	0
block3_conv1 (Conv2D)	(None, None, None, 256)	295168
block3_conv2 (Conv2D)	(None, None, None, 256)	590080
block3_conv3 (Conv2D)	(None, None, None, 256)	590080
block3_pool (MaxPooling2D)	(None, None, None, 256)	0
block4_conv1 (Conv2D)	(None, None, None, 512)	1180160
block4_conv2 (Conv2D)	(None, None, None, 512)	2359808
block4_conv3 (Conv2D)	(None, None, None, 512)	2359808
block4_pool (MaxPooling2D)	(None, None, None, 512)	0
block5_conv1 (Conv2D)	(None, None, None, 512)	2359808
block5_conv2 (Conv2D)	(None, None, None, 512)	2359808
block5_conv3 (Conv2D)	(None, None, None, 512)	2359808
block5_pool (MaxPooling2D)	(None, None, None, 512)	0

Total params: 14,714,688

Trainable params: 7,079,424

Non-trainable params: 7,635,264

Epoch 1/30

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

63/63 [=====] - 30s 444ms/step - loss: 0.1803 -

accuracy: 0.9965 - val_loss: 2.2804 - val_accuracy: 0.9810

Epoch 2/30

63/63 [=====] - 26s 417ms/step - loss: 0.0778 -

accuracy: 0.9965 - val_loss: 1.8844 - val_accuracy: 0.9840

Epoch 3/30
63/63 [=====] - 27s 423ms/step - loss: 0.0346 - accuracy: 0.9990 - val_loss: 1.7801 - val_accuracy: 0.9830

Epoch 4/30
63/63 [=====] - 27s 423ms/step - loss: 0.0174 - accuracy: 0.9975 - val_loss: 1.8610 - val_accuracy: 0.9840

Epoch 5/30
63/63 [=====] - 26s 420ms/step - loss: 0.0928 - accuracy: 0.9980 - val_loss: 2.5962 - val_accuracy: 0.9760

Epoch 6/30
63/63 [=====] - 25s 402ms/step - loss: 0.0329 - accuracy: 0.9985 - val_loss: 2.3962 - val_accuracy: 0.9800

Epoch 7/30
63/63 [=====] - 28s 452ms/step - loss: 0.1176 - accuracy: 0.9965 - val_loss: 2.2940 - val_accuracy: 0.9810

Epoch 8/30
63/63 [=====] - 27s 426ms/step - loss: 0.0327 - accuracy: 0.9975 - val_loss: 3.0157 - val_accuracy: 0.9780

Epoch 9/30
63/63 [=====] - 27s 426ms/step - loss: 0.1894 - accuracy: 0.9975 - val_loss: 2.2012 - val_accuracy: 0.9830

Epoch 10/30
63/63 [=====] - 26s 412ms/step - loss: 0.0480 - accuracy: 0.9980 - val_loss: 2.4525 - val_accuracy: 0.9800

Epoch 11/30
63/63 [=====] - 26s 409ms/step - loss: 0.1871 - accuracy: 0.9950 - val_loss: 2.7838 - val_accuracy: 0.9750

Epoch 12/30
63/63 [=====] - 26s 407ms/step - loss: 0.1754 - accuracy: 0.9970 - val_loss: 2.1835 - val_accuracy: 0.9800

Epoch 13/30
63/63 [=====] - 26s 421ms/step - loss: 0.0119 - accuracy: 0.9985 - val_loss: 2.4142 - val_accuracy: 0.9790

Epoch 14/30
63/63 [=====] - 26s 407ms/step - loss: 0.0913 - accuracy: 0.9970 - val_loss: 1.9498 - val_accuracy: 0.9840

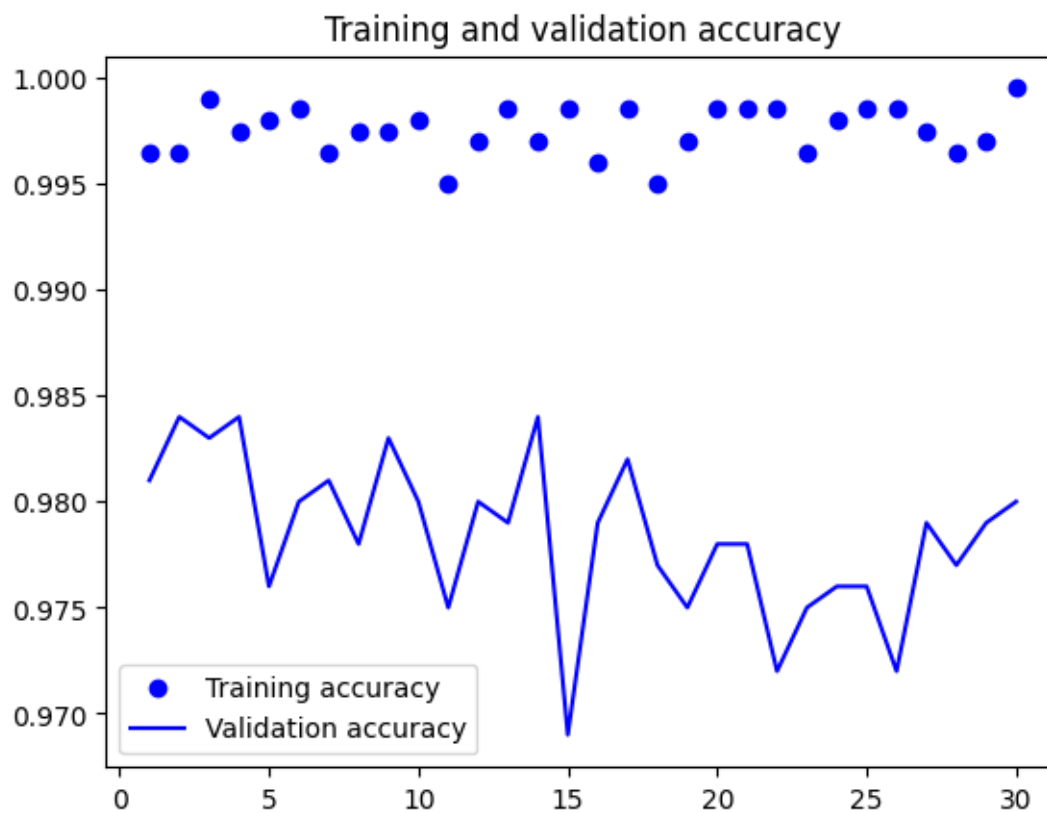
Epoch 15/30
63/63 [=====] - 28s 438ms/step - loss: 0.0740 - accuracy: 0.9985 - val_loss: 2.9215 - val_accuracy: 0.9690

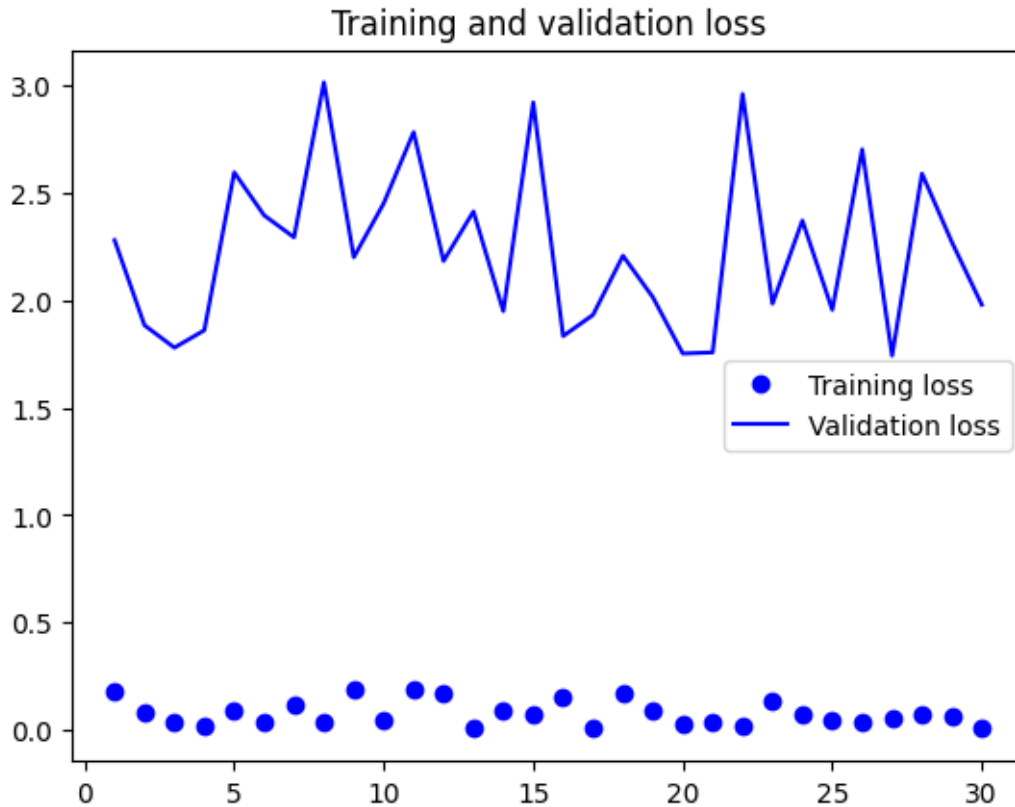
Epoch 16/30
63/63 [=====] - 25s 404ms/step - loss: 0.1489 - accuracy: 0.9960 - val_loss: 1.8340 - val_accuracy: 0.9790

Epoch 17/30
63/63 [=====] - 26s 406ms/step - loss: 0.0091 - accuracy: 0.9985 - val_loss: 1.9329 - val_accuracy: 0.9820

Epoch 18/30
63/63 [=====] - 26s 421ms/step - loss: 0.1748 - accuracy: 0.9950 - val_loss: 2.2082 - val_accuracy: 0.9770

Epoch 19/30
63/63 [=====] - 29s 460ms/step - loss: 0.0898 -
accuracy: 0.9970 - val_loss: 2.0150 - val_accuracy: 0.9750
Epoch 20/30
63/63 [=====] - 28s 439ms/step - loss: 0.0311 -
accuracy: 0.9985 - val_loss: 1.7529 - val_accuracy: 0.9780
Epoch 21/30
63/63 [=====] - 27s 428ms/step - loss: 0.0339 -
accuracy: 0.9985 - val_loss: 1.7580 - val_accuracy: 0.9780
Epoch 22/30
63/63 [=====] - 27s 430ms/step - loss: 0.0205 -
accuracy: 0.9985 - val_loss: 2.9613 - val_accuracy: 0.9720
Epoch 23/30
63/63 [=====] - 28s 439ms/step - loss: 0.1376 -
accuracy: 0.9965 - val_loss: 1.9848 - val_accuracy: 0.9750
Epoch 24/30
63/63 [=====] - 26s 417ms/step - loss: 0.0733 -
accuracy: 0.9980 - val_loss: 2.3719 - val_accuracy: 0.9760
Epoch 25/30
63/63 [=====] - 27s 425ms/step - loss: 0.0461 -
accuracy: 0.9985 - val_loss: 1.9561 - val_accuracy: 0.9760
Epoch 26/30
63/63 [=====] - 26s 414ms/step - loss: 0.0377 -
accuracy: 0.9985 - val_loss: 2.7030 - val_accuracy: 0.9720
Epoch 27/30
63/63 [=====] - 27s 431ms/step - loss: 0.0525 -
accuracy: 0.9975 - val_loss: 1.7432 - val_accuracy: 0.9790
Epoch 28/30
63/63 [=====] - 27s 425ms/step - loss: 0.0690 -
accuracy: 0.9965 - val_loss: 2.5904 - val_accuracy: 0.9770
Epoch 29/30
63/63 [=====] - 26s 408ms/step - loss: 0.0677 -
accuracy: 0.9970 - val_loss: 2.2714 - val_accuracy: 0.9790
Epoch 30/30
63/63 [=====] - 25s 396ms/step - loss: 0.0057 -
accuracy: 0.9995 - val_loss: 1.9800 - val_accuracy: 0.9800
32/32 [=====] - 6s 180ms/step - loss: 2.3243 -
accuracy: 0.9750
Test accuracy: 0.975





```
[96]: # fit-tuning with train/val/test 5000 1000 1000

original_dir = pathlib.Path('C:\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\train')
new_base_dir = pathlib.Path('C:
    ↪\\Users\\zhong\\Desktop\\FA23_DL\\HW2\\cats_vs_dogs_hw2_fit_model5')

def make_subset(subset_name, start_index, end_index):
    for category in ("cat", "dog"):
        dir = new_base_dir / subset_name / category
        os.makedirs(dir)
        fnames = [f"{category}.{i}.jpg"
                    for i in range(start_index, end_index)]
        for fname in fnames:
            shutil.copyfile(src=original_dir / fname,
                            dst=dir / fname)
make_subset("train", start_index=0, end_index=5000)          ###5000   ↵
    ↪###      5000 10000
make_subset("validation", start_index=5000, end_index=6000)  ###1000
make_subset("test", start_index=6000, end_index=7000)        ###1000 TEST
```

```

[97]: # Fine-tuning a pretrained model

conv_base.summary()

# Freezing all layers until the fourth from the last

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

# Fine-tuning the model

model.compile(loss="binary_crossentropy",
              optimizer=keras.optimizers.RMSprop(learning_rate=1e-5),
              metrics=["accuracy"])

callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="fine_tuning.keras",
        save_best_only=True,
        monitor="val_loss")
]
history = model.fit(
    train_dataset,
    epochs=30,
    validation_data=validation_dataset,
    callbacks=callbacks)
model = keras.models.load_model("fine_tuning.keras")
test_loss, test_acc = model.evaluate(test_dataset)
print(f"Test accuracy: {test_acc:.3f}")

# Plotting the results

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

```

```
plt.legend()
plt.show()
```

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_12 (InputLayer)	[(None, None, None, 3)]	0
block1_conv1 (Conv2D)	(None, None, None, 64)	1792
block1_conv2 (Conv2D)	(None, None, None, 64)	36928
block1_pool (MaxPooling2D)	(None, None, None, 64)	0
block2_conv1 (Conv2D)	(None, None, None, 128)	73856
block2_conv2 (Conv2D)	(None, None, None, 128)	147584
block2_pool (MaxPooling2D)	(None, None, None, 128)	0
block3_conv1 (Conv2D)	(None, None, None, 256)	295168
block3_conv2 (Conv2D)	(None, None, None, 256)	590080
block3_conv3 (Conv2D)	(None, None, None, 256)	590080
block3_pool (MaxPooling2D)	(None, None, None, 256)	0
block4_conv1 (Conv2D)	(None, None, None, 512)	1180160
block4_conv2 (Conv2D)	(None, None, None, 512)	2359808
block4_conv3 (Conv2D)	(None, None, None, 512)	2359808
block4_pool (MaxPooling2D)	(None, None, None, 512)	0
block5_conv1 (Conv2D)	(None, None, None, 512)	2359808
block5_conv2 (Conv2D)	(None, None, None, 512)	2359808
block5_conv3 (Conv2D)	(None, None, None, 512)	2359808
block5_pool (MaxPooling2D)	(None, None, None, 512)	0

Total params: 14,714,688

Trainable params: 7,079,424

Non-trainable params: 7,635,264

Epoch 1/30

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

63/63 [=====] - 50s 738ms/step - loss: 0.0170 -

accuracy: 0.9990 - val_loss: 1.8290 - val_accuracy: 0.9800

Epoch 2/30

63/63 [=====] - 43s 684ms/step - loss: 8.2737e-04 -

accuracy: 0.9995 - val_loss: 1.9464 - val_accuracy: 0.9800

Epoch 3/30
63/63 [=====] - 43s 687ms/step - loss: 0.0632 - accuracy: 0.9980 - val_loss: 2.0019 - val_accuracy: 0.9800

Epoch 4/30
63/63 [=====] - 44s 702ms/step - loss: 0.0549 - accuracy: 0.9960 - val_loss: 1.9564 - val_accuracy: 0.9830

Epoch 5/30
63/63 [=====] - 45s 714ms/step - loss: 0.0119 - accuracy: 0.9985 - val_loss: 2.1451 - val_accuracy: 0.9820

Epoch 6/30
63/63 [=====] - 37s 592ms/step - loss: 0.0742 - accuracy: 0.9970 - val_loss: 1.7986 - val_accuracy: 0.9850

Epoch 7/30
63/63 [=====] - 43s 692ms/step - loss: 6.0375e-04 - accuracy: 0.9995 - val_loss: 1.9648 - val_accuracy: 0.9820

Epoch 8/30
63/63 [=====] - 43s 690ms/step - loss: 0.0083 - accuracy: 0.9985 - val_loss: 2.0678 - val_accuracy: 0.9810

Epoch 9/30
63/63 [=====] - 38s 596ms/step - loss: 0.0295 - accuracy: 0.9985 - val_loss: 2.3216 - val_accuracy: 0.9800

Epoch 10/30
63/63 [=====] - 45s 719ms/step - loss: 0.0114 - accuracy: 0.9995 - val_loss: 2.2497 - val_accuracy: 0.9800

Epoch 11/30
63/63 [=====] - 45s 715ms/step - loss: 0.0358 - accuracy: 0.9985 - val_loss: 1.9221 - val_accuracy: 0.9840

Epoch 12/30
63/63 [=====] - 38s 609ms/step - loss: 0.0619 - accuracy: 0.9985 - val_loss: 2.1552 - val_accuracy: 0.9810

Epoch 13/30
63/63 [=====] - 45s 713ms/step - loss: 0.0414 - accuracy: 0.9995 - val_loss: 1.7885 - val_accuracy: 0.9840

Epoch 14/30
63/63 [=====] - 45s 713ms/step - loss: 0.1419 - accuracy: 0.9975 - val_loss: 1.8669 - val_accuracy: 0.9840

Epoch 15/30
63/63 [=====] - 39s 612ms/step - loss: 1.9374e-09 - accuracy: 1.0000 - val_loss: 1.8684 - val_accuracy: 0.9840

Epoch 16/30
63/63 [=====] - 45s 719ms/step - loss: 0.0246 - accuracy: 0.9985 - val_loss: 2.5105 - val_accuracy: 0.9750

Epoch 17/30
63/63 [=====] - 45s 716ms/step - loss: 0.0156 - accuracy: 0.9985 - val_loss: 1.5733 - val_accuracy: 0.9840

Epoch 18/30
63/63 [=====] - 38s 606ms/step - loss: 0.0017 - accuracy: 0.9990 - val_loss: 1.5487 - val_accuracy: 0.9830

Epoch 19/30
63/63 [=====] - 45s 715ms/step - loss: 0.0395 -
accuracy: 0.9980 - val_loss: 3.0671 - val_accuracy: 0.9730
Epoch 20/30
63/63 [=====] - 45s 711ms/step - loss: 0.0034 -
accuracy: 0.9995 - val_loss: 2.2679 - val_accuracy: 0.9810
Epoch 21/30
63/63 [=====] - 40s 638ms/step - loss: 0.1501 -
accuracy: 0.9985 - val_loss: 2.4323 - val_accuracy: 0.9790
Epoch 22/30
63/63 [=====] - 45s 715ms/step - loss: 0.0059 -
accuracy: 0.9995 - val_loss: 1.7983 - val_accuracy: 0.9840
Epoch 23/30
63/63 [=====] - 45s 721ms/step - loss: 0.0792 -
accuracy: 0.9975 - val_loss: 1.8469 - val_accuracy: 0.9840
Epoch 24/30
63/63 [=====] - 39s 615ms/step - loss: 0.0505 -
accuracy: 0.9990 - val_loss: 1.7808 - val_accuracy: 0.9830
Epoch 25/30
63/63 [=====] - 45s 717ms/step - loss: 0.0282 -
accuracy: 0.9980 - val_loss: 1.6320 - val_accuracy: 0.9820
Epoch 26/30
63/63 [=====] - 45s 717ms/step - loss: 1.8374e-04 -
accuracy: 1.0000 - val_loss: 1.5124 - val_accuracy: 0.9830
Epoch 27/30
63/63 [=====] - 37s 592ms/step - loss: 0.0046 -
accuracy: 0.9995 - val_loss: 2.1689 - val_accuracy: 0.9770
Epoch 28/30
63/63 [=====] - 48s 771ms/step - loss: 0.0751 -
accuracy: 0.9970 - val_loss: 1.6814 - val_accuracy: 0.9820
Epoch 29/30
63/63 [=====] - 44s 703ms/step - loss: 0.0460 -
accuracy: 0.9985 - val_loss: 1.4512 - val_accuracy: 0.9840
Epoch 30/30
63/63 [=====] - 37s 587ms/step - loss: 0.0060 -
accuracy: 0.9990 - val_loss: 1.7193 - val_accuracy: 0.9830
32/32 [=====] - 3s 83ms/step - loss: 2.0400 - accuracy:
0.9780
Test accuracy: 0.978

