In [22]:

```
import os, shutil
from keras import layers
from keras import models
from keras import optimizers
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
```

In [23]:

```
# The path to the directory where the original
# dataset was uncompressed
original_dataset_dir = './kaggle_original_data/train/train'

# The directory where we will
# store our smaller dataset
base_dir = './cats_and_dogs_small'
os.mkdir(base_dir)

# Directories for our training,
# validation and test splits
train_dir = os.path.join(base_dir, 'train')
os.mkdir(train_dir)
validation_dir = os.path.join(base_dir, 'validation')
os.mkdir(validation_dir)
test_dir = os.path.join(base_dir, 'test')
os.mkdir(test_dir)

# Directory with our training cat pictures
train_cats_dir = os.path.join(train_dir, 'cats')
os.mkdir(train_cats_dir)

# Directory with our training dog pictures
train_dogs_dir = os.path.join(train_dir, 'dogs')
os.mkdir(train_dogs_dir)

# Directory with our validation cat pictures
validation_cats_dir = os.path.join(validation_dir, 'cats')
os.mkdir(validation_cats_dir)

# Directory with our validation dog pictures
validation_dogs_dir = os.path.join(validation_dir, 'dogs')
os.mkdir(validation_dogs_dir)

# Directory with our validation cat pictures
test_cats_dir = os.path.join(test_dir, 'cats')
os.mkdir(test_cats_dir)

# Directory with our validation dog pictures
test_dogs_dir = os.path.join(test_dir, 'dogs')
os.mkdir(test_dogs_dir)

# Copy first 1000 cat images to train_cats_dir
fnames = ['cat.{}.jpg'.format(i) for i in range(1000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(train_cats_dir, fname)
    shutil.copyfile(src, dst)

# Copy next 500 cat images to validation_cats_dir
```

```
fnames = ['cat.{}.jpg'.format(i) for i in range(1000, 1500)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(validation_cats_dir, fname)
    shutil.copyfile(src, dst)

# Copy next 500 cat images to test_cats_dir
fnames = ['cat.{}.jpg'.format(i) for i in range(1500, 2000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(test_cats_dir, fname)
    shutil.copyfile(src, dst)

# Copy first 1000 dog images to train_dogs_dir
fnames = ['dog.{}.jpg'.format(i) for i in range(1000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(train_dogs_dir, fname)
    shutil.copyfile(src, dst)

# Copy next 500 dog images to validation_dogs_dir
fnames = ['dog.{}.jpg'.format(i) for i in range(1000, 1500)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(validation_dogs_dir, fname)
    shutil.copyfile(src, dst)

# Copy next 500 dog images to test_dogs_dir
fnames = ['dog.{}.jpg'.format(i) for i in range(1500, 2000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(test_dogs_dir, fname)
    shutil.copyfile(src, dst)
```

```
---------------------------------------------------------------------------
FileExistsError                           Traceback (most recent call last)
<ipython-input-23-9018e64086c2> in <module>
      6 # store our smaller dataset
      7 base_dir = './cats_and_dogs_small'
----> 8 os.mkdir(base_dir)
      9
     10 # Directories for our training,

FileExistsError: [WinError 183] Cannot create a file when that file already exists: './c
ats_and_dogs_small'
```

In [24]:

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
                        input_shape=(150, 150, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
```

In [25]:

```
model.summary()
filename = './results/model_summary_6_2_a.txt'
summary_str = []
model.summary(print_fn=lambda x: summary_str.append(x))
summary_str = '\n'.join(summary_str)
import os
if not os.path.exists('results'):
    os.makedirs('results')

# Write the summary into the file
with open(filename, 'w') as f:
    f.write(summary_str)

print(f"Model summary has been written to {filename}")
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_16 (Conv2D)           (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d_16 (MaxPooling (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_17 (MaxPooling (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 15, 15, 128)       147584
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 7, 7, 128)         0
_________________________________________________________________
flatten_4 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_8 (Dense)              (None, 512)               3211776
_________________________________________________________________
dense_9 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,453,121
Trainable params: 3,453,121
Non-trainable params: 0
_________________________________________________________________
Model summary has been written to ./results/model_summary_6_2_a.txt
```

In [26]:

```
model.compile(loss='binary_crossentropy',
              optimizer=optimizers.RMSprop(lr=1e-4),
              metrics=['acc'])
```

In [27]:

```
# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        # This is the target directory
        train_dir,
        # All images will be resized to 150x150
```

```
        target_size=(150, 150),
        batch_size=20,
        # Since we use binary_crossentropy loss, we need binary labels
        class_mode='binary')

validation_generator = test_datagen.flow_from_directory(
        validation_dir,
        target_size=(150, 150),
        batch_size=20,
        class_mode='binary')
```

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

In [28]:

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

```
data batch shape: (20, 150, 150, 3)
labels batch shape: (20,)
```

In [ ]:

```
history = model.fit_generator(
      train_generator,
      steps_per_epoch=100,
      epochs=30,
      validation_data=validation_generator,
      validation_steps=50)
```

```
Epoch 1/30
100/100 [==============================] - 33s 328ms/step - loss: 0.6919 - acc: 0.5220 -
val_loss: 0.6731 - val_acc: 0.6250
Epoch 2/30
100/100 [==============================] - 33s 328ms/step - loss: 0.6574 - acc: 0.6150 -
val_loss: 0.6530 - val_acc: 0.6120
Epoch 3/30
100/100 [==============================] - 32s 322ms/step - loss: 0.6025 - acc: 0.6650 -
val_loss: 0.6131 - val_acc: 0.6530
Epoch 4/30
100/100 [==============================] - 34s 335ms/step - loss: 0.5705 - acc: 0.7125 -
val_loss: 0.6090 - val_acc: 0.6640
Epoch 5/30
100/100 [==============================] - 34s 344ms/step - loss: 0.5418 - acc: 0.7145 -
val_loss: 0.6382 - val_acc: 0.6290
Epoch 6/30
100/100 [==============================] - 36s 359ms/step - loss: 0.5136 - acc: 0.7450 -
val_loss: 0.5826 - val_acc: 0.7030
Epoch 7/30
100/100 [==============================] - 34s 344ms/step - loss: 0.4848 - acc: 0.7635 -
val_loss: 0.5915 - val_acc: 0.6890
Epoch 8/30
100/100 [==============================] - 33s 330ms/step - loss: 0.4624 - acc: 0.7750 -
val_loss: 0.5637 - val_acc: 0.7100
Epoch 9/30
100/100 [==============================] - 32s 321ms/step - loss: 0.4402 - acc: 0.7835 -
val_loss: 0.5680 - val_acc: 0.7020
Epoch 10/30
100/100 [==============================] - 32s 324ms/step - loss: 0.4135 - acc: 0.8140 -
val_loss: 0.6012 - val_acc: 0.6980
Epoch 11/30
100/100 [==============================] - 32s 320ms/step - loss: 0.3911 - acc: 0.8300 -
val_loss: 0.5646 - val_acc: 0.7230
```

```
Epoch 12/30
100/100 [==============================] - 32s 319ms/step - loss: 0.3536 - acc: 0.8355 -
val_loss: 0.6119 - val_acc: 0.7050
Epoch 13/30
100/100 [==============================] - 32s 320ms/step - loss: 0.3361 - acc: 0.8515 -
val_loss: 0.6057 - val_acc: 0.7020
Epoch 14/30
100/100 [==============================] - 32s 320ms/step - loss: 0.3208 - acc: 0.8665 -
val_loss: 0.6480 - val_acc: 0.7160
Epoch 15/30
100/100 [==============================] - 32s 319ms/step - loss: 0.2921 - acc: 0.8790 -
val_loss: 0.5947 - val_acc: 0.7190
Epoch 16/30
100/100 [==============================] - 33s 329ms/step - loss: 0.2706 - acc: 0.8925 -
val_loss: 0.6127 - val_acc: 0.7190
Epoch 17/30
100/100 [==============================] - 33s 328ms/step - loss: 0.2411 - acc: 0.9015 -
val_loss: 0.6237 - val_acc: 0.7130
Epoch 18/30
100/100 [==============================] - 34s 336ms/step - loss: 0.2201 - acc: 0.9135 -
val_loss: 0.6718 - val_acc: 0.7080
Epoch 19/30
100/100 [==============================] - 35s 349ms/step - loss: 0.2067 - acc: 0.9180 -
val_loss: 0.6664 - val_acc: 0.7210
Epoch 20/30
100/100 [==============================] - 35s 354ms/step - loss: 0.1854 - acc: 0.9245 -
val_loss: 0.6640 - val_acc: 0.7260
Epoch 21/30
100/100 [==============================] - 34s 339ms/step - loss: 0.1683 - acc: 0.9425 -
val_loss: 0.7152 - val_acc: 0.7150
Epoch 22/30
100/100 [==============================] - 34s 337ms/step - loss: 0.1402 - acc: 0.9550 -
val_loss: 0.7589 - val_acc: 0.7090
Epoch 23/30
100/100 [==============================] - 32s 320ms/step - loss: 0.1290 - acc: 0.9585 -
val_loss: 0.8599 - val_acc: 0.7220
Epoch 24/30
100/100 [==============================] - 34s 335ms/step - loss: 0.1094 - acc: 0.9650 -
val_loss: 0.8161 - val_acc: 0.7090
Epoch 25/30
100/100 [==============================] - 34s 336ms/step - loss: 0.0923 - acc: 0.9705 -
val_loss: 0.9202 - val_acc: 0.7060
Epoch 26/30
100/100 [==============================] - 33s 328ms/step - loss: 0.0762 - acc: 0.9780 -
val_loss: 0.9165 - val_acc: 0.7090
Epoch 27/30
100/100 [==============================] - 33s 334ms/step - loss: 0.0684 - acc: 0.9800 -
val_loss: 0.9225 - val_acc: 0.7160
Epoch 28/30
100/100 [==============================] - 33s 327ms/step - loss: 0.0613 - acc: 0.9805 -
val_loss: 0.9768 - val_acc: 0.7080
Epoch 29/30
100/100 [==============================] - 33s 329ms/step - loss: 0.0587 - acc: 0.9845 -
val_loss: 0.9934 - val_acc: 0.7110
Epoch 30/30
 71/100 [====================>.........] - ETA: 8s - loss: 0.0526 - acc: 0.9866
```

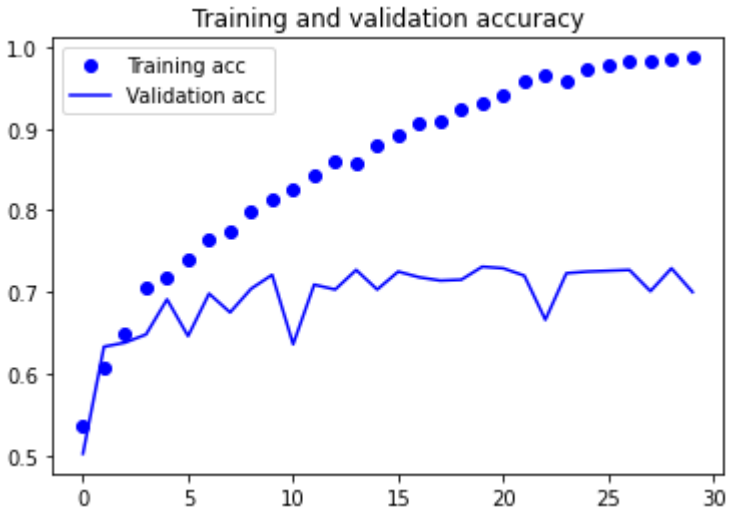
In [13]:

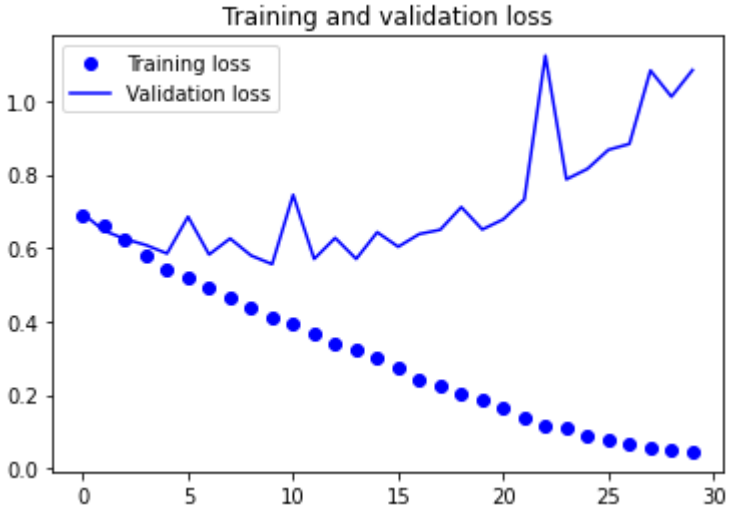
```
model.save('cats_and_dogs_small_1.h5')
```

In [14]:

```
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
```

```
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
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()
```

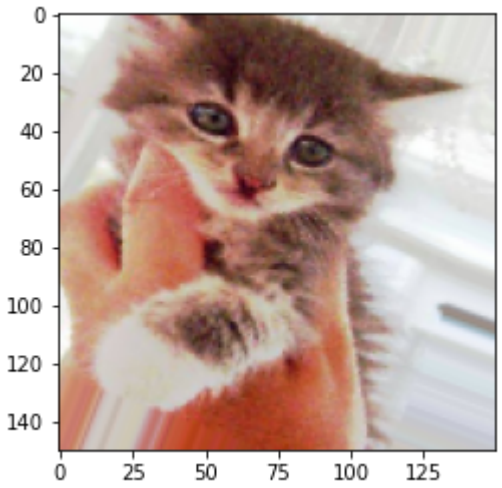


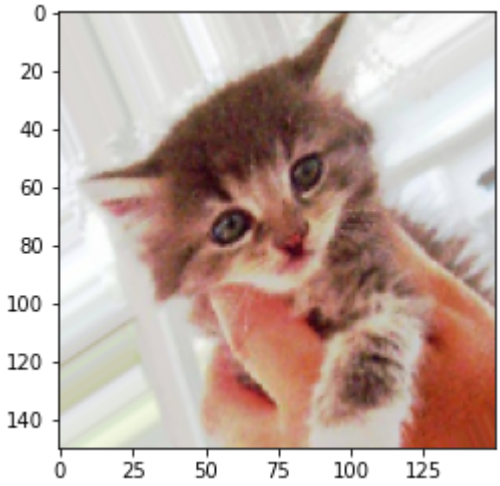


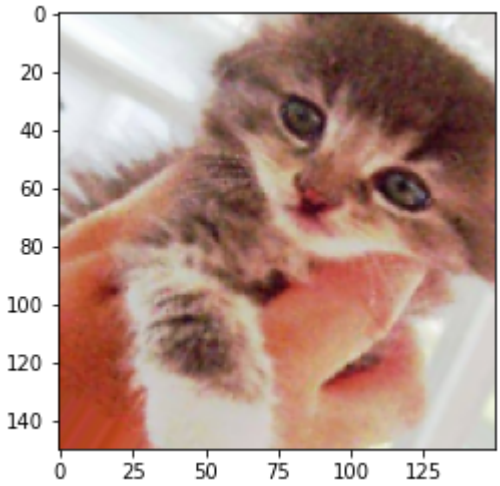
In [15]:

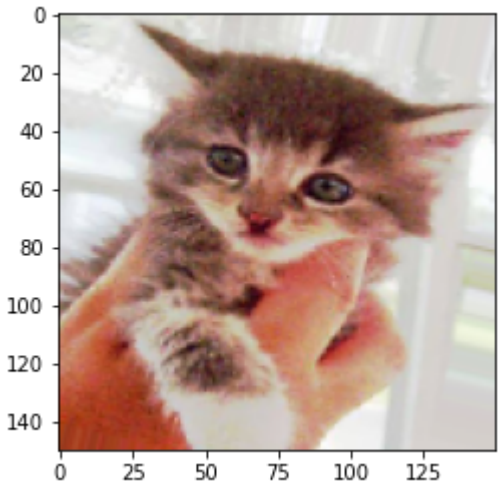
```
datagen = ImageDataGenerator(
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest')
```

```python
# This is module with image preprocessing utilities
from keras.preprocessing import image

fnames = [os.path.join(train_cats_dir, fname) for fname in os.listdir(train_cats_dir)]

# We pick one image to "augment"
img_path = fnames[3]

# Read the image and resize it
img = image.load_img(img_path, target_size=(150, 150))

# Convert it to a Numpy array with shape (150, 150, 3)
x = image.img_to_array(img)

# Reshape it to (1, 150, 150, 3)
x = x.reshape((1,) + x.shape)

# The .flow() command below generates batches of randomly transformed images.
# It will loop indefinitely, so we need to `break` the loop at some point!
i = 0
for batch in datagen.flow(x, batch_size=1):
    plt.figure(i)
    imgplot = plt.imshow(image.array_to_img(batch[0]))
    i += 1
    if i % 4 == 0:
        break

plt.show()
```









In [17]:

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',
                        input_shape=(150, 150, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
```

```
model.add(layers.Dropout(0.5))
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer=optimizers.RMSprop(lr=1e-4),
              metrics=['acc'])

train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,)

# Note that the validation data should not be augmented!
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        # This is the target directory
        train_dir,
        # All images will be resized to 150x150
        target_size=(150, 150),
        batch_size=32,
        # Since we use binary_crossentropy loss, we need binary labels
        class_mode='binary')

validation_generator = test_datagen.flow_from_directory(
        validation_dir,
        target_size=(150, 150),
        batch_size=32,
        class_mode='binary')

history = model.fit_generator(
      train_generator,
      steps_per_epoch=63,
      epochs=100,
      validation_data=validation_generator,
      validation_steps=32)


model.save('cats_and_dogs_small_2.h5')

acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
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()
```

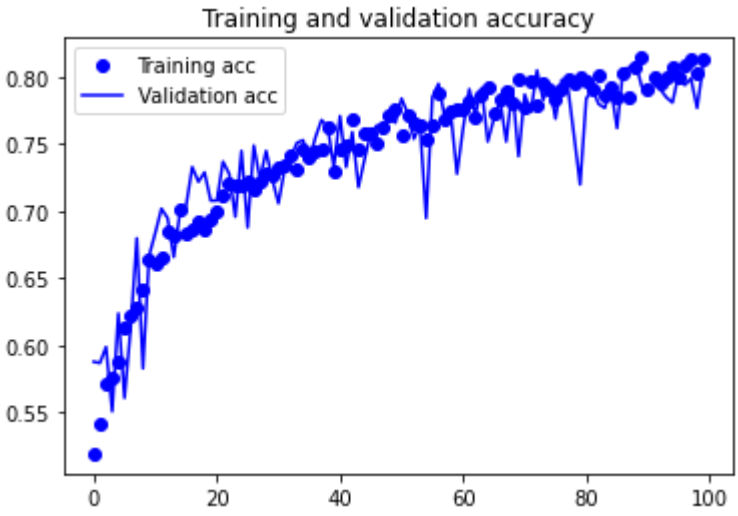
```
Found 2000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
Epoch 1/100
63/63 [==============================] - 32s 504ms/step - loss: 0.6931 - acc: 0.5195 - v
al_loss: 0.6876 - val_acc: 0.5880
Epoch 2/100
63/63 [==============================] - 32s 505ms/step - loss: 0.6889 - acc: 0.5420 - v
al_loss: 0.6782 - val_acc: 0.5870
Epoch 3/100
63/63 [==============================] - 31s 491ms/step - loss: 0.6793 - acc: 0.5710 - v
al_loss: 0.6638 - val_acc: 0.5990
Epoch 4/100
63/63 [==============================] - 31s 499ms/step - loss: 0.6707 - acc: 0.5755 - v
al_loss: 0.6764 - val_acc: 0.5510
Epoch 5/100
63/63 [==============================] - 31s 498ms/step - loss: 0.6685 - acc: 0.5875 - v
al_loss: 0.6417 - val_acc: 0.6240
Epoch 6/100
63/63 [==============================] - 31s 485ms/step - loss: 0.6580 - acc: 0.6140 - v
al_loss: 0.6695 - val_acc: 0.5610
Epoch 7/100
63/63 [==============================] - 30s 474ms/step - loss: 0.6417 - acc: 0.6225 - v
al_loss: 0.6406 - val_acc: 0.6130
Epoch 8/100
63/63 [==============================] - 30s 472ms/step - loss: 0.6382 - acc: 0.6290 - v
al_loss: 0.5973 - val_acc: 0.6800
Epoch 9/100
63/63 [==============================] - 30s 468ms/step - loss: 0.6225 - acc: 0.6415 - v
al_loss: 0.7067 - val_acc: 0.5830
Epoch 10/100
63/63 [==============================] - 30s 471ms/step - loss: 0.6104 - acc: 0.6645 - v
al_loss: 0.5877 - val_acc: 0.6670
Epoch 11/100
63/63 [==============================] - 30s 469ms/step - loss: 0.6079 - acc: 0.6615 - v
al_loss: 0.5858 - val_acc: 0.6840
Epoch 12/100
63/63 [==============================] - 30s 469ms/step - loss: 0.6059 - acc: 0.6660 - v
al_loss: 0.5618 - val_acc: 0.7020
Epoch 13/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5894 - acc: 0.6855 - v
al_loss: 0.5649 - val_acc: 0.6950
Epoch 14/100
63/63 [==============================] - 30s 473ms/step - loss: 0.5941 - acc: 0.6820 - v
al_loss: 0.6146 - val_acc: 0.6660
Epoch 15/100
63/63 [==============================] - 30s 471ms/step - loss: 0.5797 - acc: 0.7010 - v
al_loss: 0.5693 - val_acc: 0.6990
Epoch 16/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5822 - acc: 0.6835 - v
al_loss: 0.5499 - val_acc: 0.7040
Epoch 17/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5849 - acc: 0.6860 - v
al_loss: 0.5355 - val_acc: 0.7330
Epoch 18/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5787 - acc: 0.6920 - v
al_loss: 0.5433 - val_acc: 0.7220
Epoch 19/100
63/63 [==============================] - 30s 472ms/step - loss: 0.5752 - acc: 0.6865 - v
al_loss: 0.5382 - val_acc: 0.7290
```

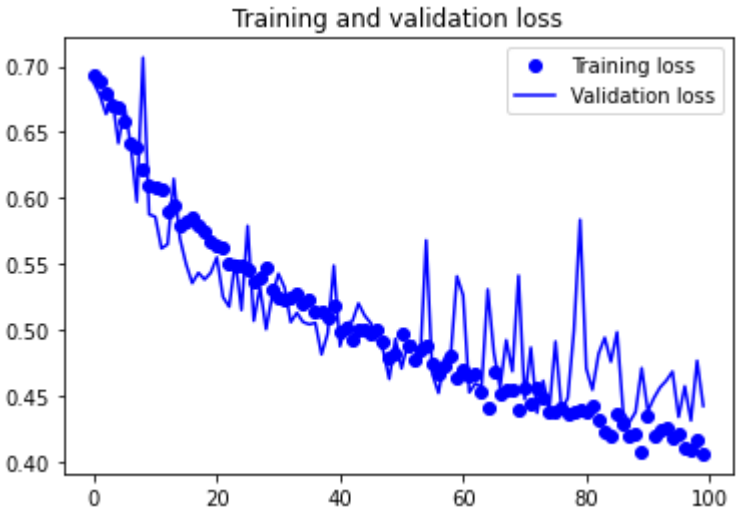
```
Epoch 20/100
63/63 [==============================] - 30s 469ms/step - loss: 0.5672 - acc: 0.6940 - v
al_loss: 0.5433 - val_acc: 0.7080
Epoch 21/100
63/63 [==============================] - 30s 472ms/step - loss: 0.5634 - acc: 0.7005 - v
al_loss: 0.5550 - val_acc: 0.7080
Epoch 22/100
63/63 [==============================] - 30s 469ms/step - loss: 0.5633 - acc: 0.7115 - v
al_loss: 0.5249 - val_acc: 0.7370
Epoch 23/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5508 - acc: 0.7205 - v
al_loss: 0.5175 - val_acc: 0.7280
Epoch 24/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5485 - acc: 0.7200 - v
al_loss: 0.5527 - val_acc: 0.6960
Epoch 25/100
63/63 [==============================] - 30s 469ms/step - loss: 0.5490 - acc: 0.7200 - v
al_loss: 0.5150 - val_acc: 0.7450
Epoch 26/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5459 - acc: 0.7230 - v
al_loss: 0.5789 - val_acc: 0.6880
Epoch 27/100
63/63 [==============================] - 30s 473ms/step - loss: 0.5363 - acc: 0.7165 - v
al_loss: 0.5068 - val_acc: 0.7490
Epoch 28/100
63/63 [==============================] - 30s 471ms/step - loss: 0.5395 - acc: 0.7220 - v
al_loss: 0.5325 - val_acc: 0.7240
Epoch 29/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5470 - acc: 0.7285 - v
al_loss: 0.5004 - val_acc: 0.7450
Epoch 30/100
63/63 [==============================] - 30s 476ms/step - loss: 0.5302 - acc: 0.7270 - v
al_loss: 0.5263 - val_acc: 0.7240
Epoch 31/100
63/63 [==============================] - 30s 471ms/step - loss: 0.5241 - acc: 0.7330 - v
al_loss: 0.5426 - val_acc: 0.7060
Epoch 32/100
63/63 [==============================] - 30s 471ms/step - loss: 0.5225 - acc: 0.7350 - v
al_loss: 0.5321 - val_acc: 0.7310
Epoch 33/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5250 - acc: 0.7415 - v
al_loss: 0.5059 - val_acc: 0.7400
Epoch 34/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5274 - acc: 0.7310 - v
al_loss: 0.5129 - val_acc: 0.7510
Epoch 35/100
63/63 [==============================] - 30s 471ms/step - loss: 0.5200 - acc: 0.7455 - v
al_loss: 0.5059 - val_acc: 0.7530
Epoch 36/100
63/63 [==============================] - 30s 471ms/step - loss: 0.5237 - acc: 0.7400 - v
al_loss: 0.5042 - val_acc: 0.7380
Epoch 37/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5137 - acc: 0.7450 - v
al_loss: 0.5053 - val_acc: 0.7550
Epoch 38/100
63/63 [==============================] - 30s 476ms/step - loss: 0.5138 - acc: 0.7455 - v
al_loss: 0.4814 - val_acc: 0.7680
Epoch 39/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5088 - acc: 0.7625 - v
al_loss: 0.4973 - val_acc: 0.7630
Epoch 40/100
63/63 [==============================] - 30s 469ms/step - loss: 0.5186 - acc: 0.7295 - v
al_loss: 0.5488 - val_acc: 0.7330
Epoch 41/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4982 - acc: 0.7465 - v
```

```
al_loss: 0.4876 - val_acc: 0.7710
Epoch 42/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5020 - acc: 0.7495 - v
al_loss: 0.5050 - val_acc: 0.7330
Epoch 43/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4930 - acc: 0.7685 - v
al_loss: 0.5063 - val_acc: 0.7590
Epoch 44/100
63/63 [==============================] - 30s 469ms/step - loss: 0.5008 - acc: 0.7455 - v
al_loss: 0.5201 - val_acc: 0.7180
Epoch 45/100
63/63 [==============================] - 30s 475ms/step - loss: 0.5000 - acc: 0.7580 - v
al_loss: 0.5109 - val_acc: 0.7390
Epoch 46/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4975 - acc: 0.7575 - v
al_loss: 0.5051 - val_acc: 0.7580
Epoch 47/100
63/63 [==============================] - 30s 470ms/step - loss: 0.5000 - acc: 0.7505 - v
al_loss: 0.4915 - val_acc: 0.7640
Epoch 48/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4916 - acc: 0.7630 - v
al_loss: 0.4847 - val_acc: 0.7580
Epoch 49/100
63/63 [==============================] - 30s 472ms/step - loss: 0.4791 - acc: 0.7720 - v
al_loss: 0.4627 - val_acc: 0.7700
Epoch 50/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4819 - acc: 0.7755 - v
al_loss: 0.4932 - val_acc: 0.7660
Epoch 51/100
63/63 [==============================] - 31s 485ms/step - loss: 0.4971 - acc: 0.7565 - v
al_loss: 0.4707 - val_acc: 0.7840
Epoch 52/100
63/63 [==============================] - 31s 497ms/step - loss: 0.4876 - acc: 0.7720 - v
al_loss: 0.4892 - val_acc: 0.7730
Epoch 53/100
63/63 [==============================] - 30s 484ms/step - loss: 0.4779 - acc: 0.7660 - v
al_loss: 0.4912 - val_acc: 0.7540
Epoch 54/100
63/63 [==============================] - 31s 486ms/step - loss: 0.4843 - acc: 0.7640 - v
al_loss: 0.4750 - val_acc: 0.7640
Epoch 55/100
63/63 [==============================] - 30s 481ms/step - loss: 0.4882 - acc: 0.7540 - v
al_loss: 0.5680 - val_acc: 0.6950
Epoch 56/100
63/63 [==============================] - 32s 502ms/step - loss: 0.4740 - acc: 0.7640 - v
al_loss: 0.4666 - val_acc: 0.7850
Epoch 57/100
63/63 [==============================] - 33s 521ms/step - loss: 0.4664 - acc: 0.7875 - v
al_loss: 0.4520 - val_acc: 0.7950
Epoch 58/100
63/63 [==============================] - 33s 525ms/step - loss: 0.4732 - acc: 0.7680 - v
al_loss: 0.4787 - val_acc: 0.7700
Epoch 59/100
63/63 [==============================] - 31s 491ms/step - loss: 0.4811 - acc: 0.7740 - v
al_loss: 0.4813 - val_acc: 0.7730
Epoch 60/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4642 - acc: 0.7760 - v
al_loss: 0.5406 - val_acc: 0.7280
Epoch 61/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4701 - acc: 0.7755 - v
al_loss: 0.5262 - val_acc: 0.7620
Epoch 62/100
63/63 [==============================] - 29s 468ms/step - loss: 0.4657 - acc: 0.7820 - v
al_loss: 0.4525 - val_acc: 0.7910
Epoch 63/100
```

```
63/63 [==============================] - 30s 473ms/step - loss: 0.4663 - acc: 0.7695 - v
al_loss: 0.4593 - val_acc: 0.7720
Epoch 64/100
63/63 [==============================] - 30s 471ms/step - loss: 0.4534 - acc: 0.7870 - v
al_loss: 0.4567 - val_acc: 0.7920
Epoch 65/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4404 - acc: 0.7925 - v
al_loss: 0.5307 - val_acc: 0.7520
Epoch 66/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4676 - acc: 0.7730 - v
al_loss: 0.4826 - val_acc: 0.7690
Epoch 67/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4518 - acc: 0.7840 - v
al_loss: 0.4559 - val_acc: 0.7830
Epoch 68/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4547 - acc: 0.7890 - v
al_loss: 0.4922 - val_acc: 0.7520
Epoch 69/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4551 - acc: 0.7810 - v
al_loss: 0.4686 - val_acc: 0.7920
Epoch 70/100
63/63 [==============================] - 30s 473ms/step - loss: 0.4389 - acc: 0.7990 - v
al_loss: 0.5412 - val_acc: 0.7410
Epoch 71/100
63/63 [==============================] - 29s 468ms/step - loss: 0.4561 - acc: 0.7770 - v
al_loss: 0.4474 - val_acc: 0.7870
Epoch 72/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4432 - acc: 0.7970 - v
al_loss: 0.4866 - val_acc: 0.7740
Epoch 73/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4566 - acc: 0.7790 - v
al_loss: 0.4370 - val_acc: 0.8050
Epoch 74/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4477 - acc: 0.7955 - v
al_loss: 0.4613 - val_acc: 0.7830
Epoch 75/100
63/63 [==============================] - 30s 471ms/step - loss: 0.4376 - acc: 0.7910 - v
al_loss: 0.4393 - val_acc: 0.7950
Epoch 76/100
63/63 [==============================] - 30s 471ms/step - loss: 0.4381 - acc: 0.7835 - v
al_loss: 0.4911 - val_acc: 0.7690
Epoch 77/100
63/63 [==============================] - 30s 471ms/step - loss: 0.4405 - acc: 0.7915 - v
al_loss: 0.4387 - val_acc: 0.7980
Epoch 78/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4367 - acc: 0.7980 - v
al_loss: 0.4483 - val_acc: 0.7930
Epoch 79/100
63/63 [==============================] - 30s 473ms/step - loss: 0.4375 - acc: 0.7950 - v
al_loss: 0.4994 - val_acc: 0.7570
Epoch 80/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4387 - acc: 0.7995 - v
al_loss: 0.5836 - val_acc: 0.7200
Epoch 81/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4371 - acc: 0.7970 - v
al_loss: 0.4707 - val_acc: 0.7840
Epoch 82/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4428 - acc: 0.7915 - v
al_loss: 0.4545 - val_acc: 0.7910
Epoch 83/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4315 - acc: 0.8010 - v
al_loss: 0.4818 - val_acc: 0.7800
Epoch 84/100
63/63 [==============================] - 30s 468ms/step - loss: 0.4230 - acc: 0.7865 - v
al_loss: 0.4941 - val_acc: 0.7770
```

```
Epoch 85/100
63/63 [==============================] - 30s 474ms/step - loss: 0.4195 - acc: 0.7920 - v
al_loss: 0.4758 - val_acc: 0.7920
Epoch 86/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4356 - acc: 0.7845 - v
al_loss: 0.4982 - val_acc: 0.7620
Epoch 87/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4291 - acc: 0.8035 - v
al_loss: 0.4373 - val_acc: 0.8060
Epoch 88/100
63/63 [==============================] - 30s 472ms/step - loss: 0.4203 - acc: 0.7845 - v
al_loss: 0.4298 - val_acc: 0.8070
Epoch 89/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4209 - acc: 0.8080 - v
al_loss: 0.4376 - val_acc: 0.8110
Epoch 90/100
63/63 [==============================] - 30s 473ms/step - loss: 0.4066 - acc: 0.8145 - v
al_loss: 0.4709 - val_acc: 0.7930
Epoch 91/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4352 - acc: 0.7910 - v
al_loss: 0.4382 - val_acc: 0.7890
Epoch 92/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4192 - acc: 0.8000 - v
al_loss: 0.4484 - val_acc: 0.7950
Epoch 93/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4244 - acc: 0.7960 - v
al_loss: 0.4564 - val_acc: 0.7910
Epoch 94/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4263 - acc: 0.7995 - v
al_loss: 0.4615 - val_acc: 0.7850
Epoch 95/100
63/63 [==============================] - 30s 471ms/step - loss: 0.4173 - acc: 0.8080 - v
al_loss: 0.4683 - val_acc: 0.7810
Epoch 96/100
63/63 [==============================] - 30s 475ms/step - loss: 0.4211 - acc: 0.7995 - v
al_loss: 0.4339 - val_acc: 0.8020
Epoch 97/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4107 - acc: 0.8095 - v
al_loss: 0.4572 - val_acc: 0.7940
Epoch 98/100
63/63 [==============================] - 30s 477ms/step - loss: 0.4091 - acc: 0.8135 - v
al_loss: 0.4312 - val_acc: 0.7990
Epoch 99/100
63/63 [==============================] - 30s 469ms/step - loss: 0.4163 - acc: 0.8030 - v
al_loss: 0.4765 - val_acc: 0.7770
Epoch 100/100
63/63 [==============================] - 30s 470ms/step - loss: 0.4060 - acc: 0.8130 - v
al_loss: 0.4423 - val_acc: 0.8090
```





In [32]:

```
predictions = model.predict(validation_generator)

# Convert the predictions to a string
predictions_str = '\n'.join(map(str, predictions.flatten()))

# Specify the subdirectory and the filename
filename_predictions = './results/model_predictions_6_2_b.txt'

# Write the predictions into the file
with open(filename_predictions, 'w') as f:
    f.write(predictions_str)

print(f"Model predictions have been written to {filename_predictions}")
```

```
Model predictions have been written to ./results/model_predictions_6_2_b.txt
```

In [ ]: