

In [4]:

```
import os, shutil
```

In [5]:

```
# hide warnings
import warnings
warnings.filterwarnings('ignore')
```

In [6]:

```
# The path to the directory where the original
# dataset was uncompressed
original_dataset_dir_cats = 'C:\\kiran\\datasets\\archive (1)\\dog vs cat\\dataset\\training_set\\cats'
original_dataset_dir_dogs = 'C:\\kiran\\datasets\\archive (1)\\dog vs cat\\dataset\\training_set\\dogs'
```

In [7]:

```
# The directory where we will
# store our smaller dataset
base_dir = 'C:\\kiran\\datasets\\cats_and_dogs_small'
```

In [4]:

```
os.mkdir(base_dir)
```

In [8]:

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

validation_dir = os.path.join(base_dir, 'validation')
test_dir = os.path.join(base_dir, 'test')
```

In []:

```
os.mkdir(validation_dir)
os.mkdir(train_dir)
os.mkdir(test_dir)
```

In [11]:

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

validation_dogs_dir = os.path.join(validation_dir, 'dogs')
validation_cats_dir = os.path.join(validation_dir, 'cats')

test_cats_dir = os.path.join(test_dir, 'cats')
test_dogs_dir = os.path.join(test_dir, 'dogs')
```

In []:

```
os.mkdir(train_cats_dir)

# Directory with our training dog pictures

os.mkdir(train_dogs_dir)

# Directory with our validation cat pictures

os.mkdir(validation_cats_dir)

# Directory with our validation dog pictures

os.mkdir(validation_dogs_dir)

# Directory with our validation cat pictures

os.mkdir(test_cats_dir)

# Directory with our validation dog pictures

os.mkdir(test_dogs_dir)
```

In [12]:

```
# 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_cats, 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_cats, 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_cats, 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_dogs, 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_dogs, 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_dogs, fname)
    dst = os.path.join(test_dogs_dir, fname)
    shutil.copyfile(src, dst)
```

In [12]:

```
print('total training cat images:', len(os.listdir(train_cats_dir)))
```

total training cat images: 1000

In [13]:

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

In [14]:

```

from tensorflow.keras import layers
from tensorflow.keras import models

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 [15]:

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
conv2d_3 (Conv2D)	(None, 15, 15, 128)	147584
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 128)	0
flatten (Flatten)	(None, 6272)	0
dense (Dense)	(None, 512)	3211776
dense_1 (Dense)	(None, 1)	513
Total params: 3,453,121		
Trainable params: 3,453,121		
Non-trainable params: 0		

In [17]:

```
from tensorflow.keras import optimizers

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

In [19]:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# 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 [20]:

```
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 [21]:

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

WARNING:tensorflow:From <ipython-input-21-a7acfc8093a4>:1: Model.fit_generator (from tensorflow.python.keras.engine.training) is deprecated and will be removed in a future version.

Instructions for updating:

Please use Model.fit, which supports generators.

Epoch 1/30

100/100 [=====] - 85s 851ms/step - loss: 0.6902 - acc: 0.5145 - val_loss: 0.6769 - val_acc: 0.6300

Epoch 2/30

100/100 [=====] - 77s 773ms/step - loss: 0.6583 - acc: 0.6105 - val_loss: 0.6631 - val_acc: 0.5770

Epoch 3/30

100/100 [=====] - 91s 906ms/step - loss: 0.6205 - acc: 0.6575 - val_loss: 0.5987 - val_acc: 0.6750

Epoch 4/30

100/100 [=====] - 102s 1s/step - loss: 0.5664 - acc: 0.7015 - val_loss: 0.6114 - val_acc: 0.6480

Epoch 5/30

100/100 [=====] - 104s 1s/step - loss: 0.5390 - acc: 0.7200 - val_loss: 0.5695 - val_acc: 0.6870

Epoch 6/30

100/100 [=====] - 104s 1s/step - loss: 0.5064 - acc: 0.7570 - val_loss: 0.5797 - val_acc: 0.6950

Epoch 7/30

100/100 [=====] - 104s 1s/step - loss: 0.4841 - acc: 0.7615 - val_loss: 0.5492 - val_acc: 0.7050

Epoch 8/30

100/100 [=====] - 114s 1s/step - loss: 0.4647 - acc: 0.7905 - val_loss: 0.5479 - val_acc: 0.7140

Epoch 9/30

100/100 [=====] - 111s 1s/step - loss: 0.4353 - acc: 0.7990 - val_loss: 0.5241 - val_acc: 0.7430

Epoch 10/30

100/100 [=====] - 112s 1s/step - loss: 0.4132 - acc: 0.8075 - val_loss: 0.5301 - val_acc: 0.7260

Epoch 11/30

100/100 [=====] - 113s 1s/step - loss: 0.3928 - acc: 0.8135 - val_loss: 0.5091 - val_acc: 0.7500

Epoch 12/30

100/100 [=====] - 120s 1s/step - loss: 0.3688 - acc: 0.8385 - val_loss: 0.5786 - val_acc: 0.7450

Epoch 13/30

100/100 [=====] - 113s 1s/step - loss: 0.3460 - acc: 0.8450 - val_loss: 0.5532 - val_acc: 0.7380

Epoch 14/30

100/100 [=====] - 120s 1s/step - loss: 0.3216 - acc: 0.8630 - val_loss: 0.5342 - val_acc: 0.7480

Epoch 15/30

100/100 [=====] - 119s 1s/step - loss: 0.3125 - acc: 0.8705 - val_loss: 0.5763 - val_acc: 0.7410

Epoch 16/30

100/100 [=====] - 115s 1s/step - loss: 0.2844 - acc: 0.8860 - val_loss: 0.5523 - val_acc: 0.7460

Epoch 17/30

100/100 [=====] - 116s 1s/step - loss: 0.2598 - acc: 0.8955 - val_loss: 0.5819 - val_acc: 0.7320

Epoch 18/30

100/100 [=====] - 118s 1s/step - loss: 0.2436 - acc: 0.9065 - val_loss: 0.5683 - val_acc: 0.7470

Epoch 19/30

100/100 [=====] - 118s 1s/step - loss: 0.2257 - a

```
cc: 0.9075 - val_loss: 0.6734 - val_acc: 0.7150
Epoch 20/30
100/100 [=====] - 123s 1s/step - loss: 0.2036 - a
cc: 0.9235 - val_loss: 0.6078 - val_acc: 0.7500
Epoch 21/30
100/100 [=====] - 124s 1s/step - loss: 0.1854 - a
cc: 0.9325 - val_loss: 0.7116 - val_acc: 0.7330
Epoch 22/30
100/100 [=====] - 120s 1s/step - loss: 0.1686 - a
cc: 0.9375 - val_loss: 0.6207 - val_acc: 0.7430
Epoch 23/30
100/100 [=====] - 121s 1s/step - loss: 0.1529 - a
cc: 0.9510 - val_loss: 0.8177 - val_acc: 0.7290
Epoch 24/30
100/100 [=====] - 120s 1s/step - loss: 0.1398 - a
cc: 0.9540 - val_loss: 0.6860 - val_acc: 0.7400
Epoch 25/30
100/100 [=====] - 122s 1s/step - loss: 0.1256 - a
cc: 0.9580 - val_loss: 0.6966 - val_acc: 0.7540
Epoch 26/30
100/100 [=====] - 123s 1s/step - loss: 0.1174 - a
cc: 0.9635 - val_loss: 0.7545 - val_acc: 0.7560
Epoch 27/30
100/100 [=====] - 118s 1s/step - loss: 0.1057 - a
cc: 0.9670 - val_loss: 0.7561 - val_acc: 0.7470
Epoch 28/30
100/100 [=====] - 121s 1s/step - loss: 0.0850 - a
cc: 0.9775 - val_loss: 0.8050 - val_acc: 0.7290
Epoch 29/30
100/100 [=====] - 119s 1s/step - loss: 0.0767 - a
cc: 0.9770 - val_loss: 0.7804 - val_acc: 0.7380
Epoch 30/30
100/100 [=====] - 120s 1s/step - loss: 0.0616 - a
cc: 0.9855 - val_loss: 0.8799 - val_acc: 0.7370
```

In [22]:

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


In [23]:

```
import matplotlib.pyplot as plt

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

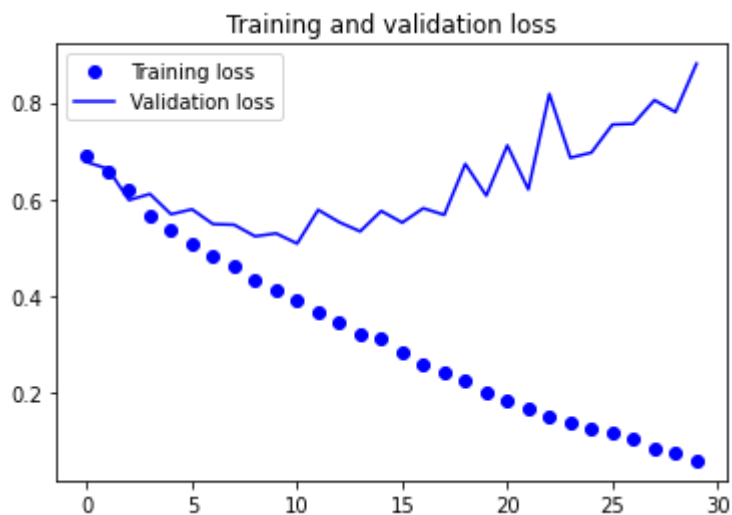
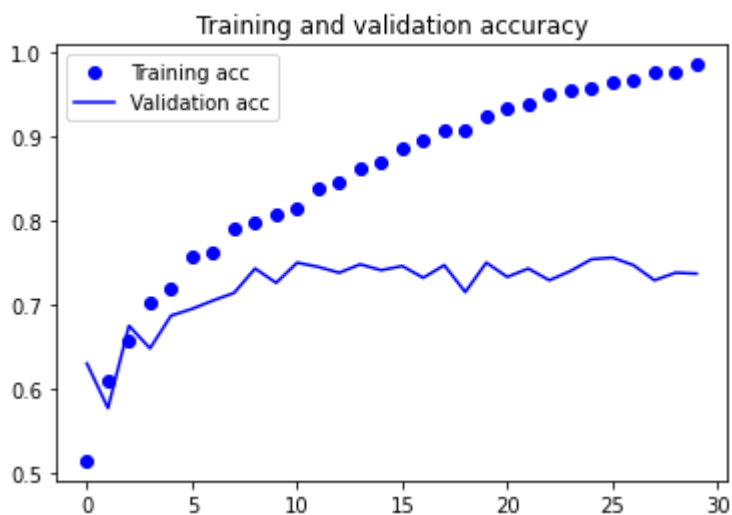
epochs = range(len(acc))

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()
```



Data Augmentation

In [24]:

```
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')
```

augmented images:

In [25]:

```
# This is module with image preprocessing utilities
from tensorflow.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]

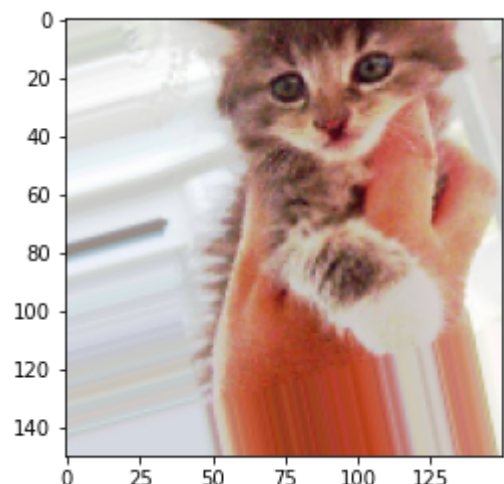
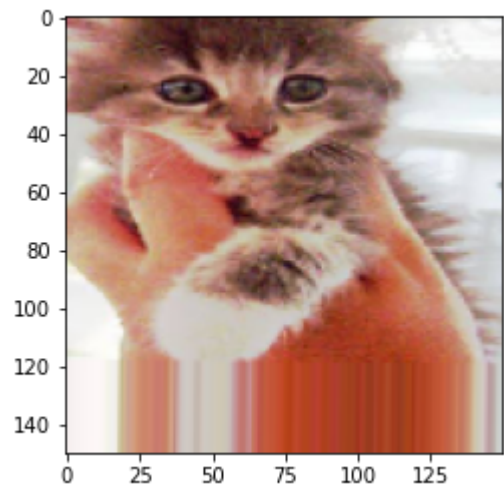
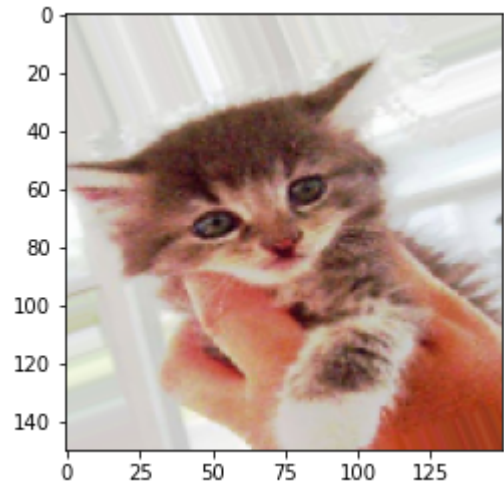
# Read the image and resize it
img = image.load_img(img_path, target_size=(150, 150))

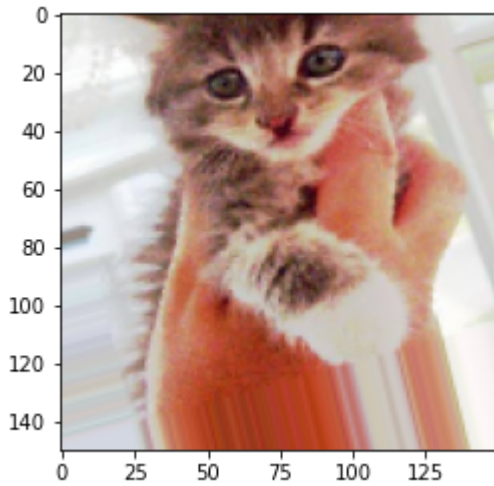
# Convert it to a Numpy array with shape (150, 150, 3)
x = image.img_to_array(img)

# 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 [26]:

```
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'])
```

In [30]:

```
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=60,
    epochs=100,
    validation_data=validation_generator,
    validation_steps=50)
```

Found 2000 images belonging to 2 classes.

Found 1000 images belonging to 2 classes.

Epoch 1/100

60/60 [=====] - ETA: 0s - loss: 0.6769 - acc: 0.5

688WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` batches (in this case, 50 batches). You may need to use the repeat() function when building your dataset.

60/60 [=====] - 81s 1s/step - loss: 0.6769 - acc: 0.5688 - val_loss: 0.6565 - val_acc: 0.5880

Epoch 2/100

60/60 [=====] - 66s 1s/step - loss: 0.6701 - acc: 0.5730

Epoch 3/100

60/60 [=====] - 74s 1s/step - loss: 0.6648 - acc: 0.5909

Epoch 4/100

60/60 [=====] - 83s 1s/step - loss: 0.6578 - acc: 0.5951

Epoch 5/100

60/60 [=====] - 83s 1s/step - loss: 0.6534 - acc: 0.6108

Epoch 6/100

60/60 [=====] - 85s 1s/step - loss: 0.6352 - acc: 0.6455

Epoch 7/100

60/60 [=====] - 84s 1s/step - loss: 0.6318 - acc: 0.6282

Epoch 8/100

60/60 [=====] - 90s 1s/step - loss: 0.6231 - acc: 0.6360

Epoch 9/100

60/60 [=====] - 91s 2s/step - loss: 0.6127 - acc: 0.6576

Epoch 10/100

60/60 [=====] - 98s 2s/step - loss: 0.6080 - acc: 0.6528

Epoch 11/100

60/60 [=====] - 94s 2s/step - loss: 0.6039 - acc: 0.6723

Epoch 12/100

60/60 [=====] - 92s 2s/step - loss: 0.5961 - acc: 0.6843

Epoch 13/100

60/60 [=====] - 97s 2s/step - loss: 0.5954 - acc: 0.6728

Epoch 14/100

60/60 [=====] - 99s 2s/step - loss: 0.5835 - acc: 0.6891

Epoch 15/100

60/60 [=====] - 100s 2s/step - loss: 0.5858 - acc: 0.6901

Epoch 16/100

60/60 [=====] - 102s 2s/step - loss: 0.5820 - acc: 0.6917

Epoch 17/100

60/60 [=====] - 100s 2s/step - loss: 0.5770 - acc: 0.6901

Epoch 18/100

60/60 [=====] - 102s 2s/step - loss: 0.5714 - acc: 0.6812

Epoch 19/100
60/60 [=====] - 106s 2s/step - loss: 0.5655 - acc: 0.7127

Epoch 20/100
60/60 [=====] - 101s 2s/step - loss: 0.5670 - acc: 0.7043

Epoch 21/100
60/60 [=====] - 98s 2s/step - loss: 0.5592 - acc: 0.7054

Epoch 22/100
60/60 [=====] - 103s 2s/step - loss: 0.5555 - acc: 0.7096

Epoch 23/100
60/60 [=====] - 102s 2s/step - loss: 0.5620 - acc: 0.7174

Epoch 24/100
60/60 [=====] - 103s 2s/step - loss: 0.5495 - acc: 0.7159

Epoch 25/100
60/60 [=====] - 101s 2s/step - loss: 0.5531 - acc: 0.7211

Epoch 26/100
60/60 [=====] - 101s 2s/step - loss: 0.5503 - acc: 0.7127

Epoch 27/100
60/60 [=====] - 103s 2s/step - loss: 0.5381 - acc: 0.7237

Epoch 28/100
60/60 [=====] - 96s 2s/step - loss: 0.5297 - acc: 0.7379

Epoch 29/100
60/60 [=====] - 101s 2s/step - loss: 0.5506 - acc: 0.7090

Epoch 30/100
60/60 [=====] - 105s 2s/step - loss: 0.5357 - acc: 0.7332

Epoch 31/100
60/60 [=====] - 1495s 25s/step - loss: 0.5426 - acc: 0.7243

Epoch 32/100
60/60 [=====] - 68s 1s/step - loss: 0.5215 - acc: 0.7348

Epoch 33/100
60/60 [=====] - 73s 1s/step - loss: 0.5215 - acc: 0.7390

Epoch 34/100
60/60 [=====] - 72s 1s/step - loss: 0.5198 - acc: 0.7416

Epoch 35/100
60/60 [=====] - 69s 1s/step - loss: 0.5133 - acc: 0.7458

Epoch 36/100
60/60 [=====] - 82s 1s/step - loss: 0.5177 - acc: 0.7405

Epoch 37/100
60/60 [=====] - 87s 1s/step - loss: 0.5163 - acc: 0.7521

Epoch 38/100
60/60 [=====] - 88s 1s/step - loss: 0.4948 - acc: 0.7679

Epoch 39/100


```
60/60 [=====] - 95s 2s/step - loss: 0.5108 - acc: 0.7484
Epoch 40/100
60/60 [=====] - 106s 2s/step - loss: 0.5011 - acc: 0.7568
Epoch 41/100
60/60 [=====] - 110s 2s/step - loss: 0.5063 - acc: 0.7437
Epoch 42/100
60/60 [=====] - 105s 2s/step - loss: 0.4992 - acc: 0.7532
Epoch 43/100
60/60 [=====] - 112s 2s/step - loss: 0.4950 - acc: 0.7600
Epoch 44/100
60/60 [=====] - 113s 2s/step - loss: 0.4993 - acc: 0.7558
Epoch 45/100
60/60 [=====] - 114s 2s/step - loss: 0.5005 - acc: 0.7547
Epoch 46/100
60/60 [=====] - 116s 2s/step - loss: 0.4795 - acc: 0.7794
Epoch 47/100
60/60 [=====] - 118s 2s/step - loss: 0.4985 - acc: 0.7574
Epoch 48/100
60/60 [=====] - 123s 2s/step - loss: 0.4952 - acc: 0.7631
Epoch 49/100
60/60 [=====] - 123s 2s/step - loss: 0.4963 - acc: 0.7626
Epoch 50/100
60/60 [=====] - 119s 2s/step - loss: 0.4839 - acc: 0.7731
Epoch 51/100
60/60 [=====] - 125s 2s/step - loss: 0.4885 - acc: 0.7663
Epoch 52/100
60/60 [=====] - 119s 2s/step - loss: 0.4919 - acc: 0.7642
Epoch 53/100
60/60 [=====] - 128s 2s/step - loss: 0.4837 - acc: 0.7700
Epoch 54/100
60/60 [=====] - 128s 2s/step - loss: 0.4635 - acc: 0.7773
Epoch 55/100
60/60 [=====] - 132s 2s/step - loss: 0.4891 - acc: 0.7568
Epoch 56/100
60/60 [=====] - 138s 2s/step - loss: 0.4709 - acc: 0.7705
Epoch 57/100
60/60 [=====] - 130s 2s/step - loss: 0.4633 - acc: 0.7794
Epoch 58/100
60/60 [=====] - 131s 2s/step - loss: 0.4679 - acc: 0.7784
Epoch 59/100
60/60 [=====] - 131s 2s/step - loss: 0.4687 - acc:
```

```
c: 0.7705
Epoch 60/100
60/60 [=====] - 134s 2s/step - loss: 0.4645 - ac
c: 0.7784
Epoch 61/100
60/60 [=====] - 134s 2s/step - loss: 0.4669 - ac
c: 0.7652
Epoch 62/100
60/60 [=====] - 138s 2s/step - loss: 0.4582 - ac
c: 0.7763
Epoch 63/100
60/60 [=====] - 125s 2s/step - loss: 0.4524 - ac
c: 0.7931
Epoch 64/100
60/60 [=====] - 127s 2s/step - loss: 0.4504 - ac
c: 0.7899
Epoch 65/100
60/60 [=====] - 129s 2s/step - loss: 0.4536 - ac
c: 0.7815
Epoch 66/100
60/60 [=====] - 135s 2s/step - loss: 0.4744 - ac
c: 0.7784
Epoch 67/100
60/60 [=====] - 138s 2s/step - loss: 0.4602 - ac
c: 0.7852
Epoch 68/100
60/60 [=====] - 135s 2s/step - loss: 0.4577 - ac
c: 0.7778
Epoch 69/100
60/60 [=====] - 137s 2s/step - loss: 0.4475 - ac
c: 0.7978
Epoch 70/100
60/60 [=====] - 136s 2s/step - loss: 0.4479 - ac
c: 0.7973
Epoch 71/100
60/60 [=====] - 122s 2s/step - loss: 0.4325 - ac
c: 0.8020
Epoch 72/100
60/60 [=====] - 122s 2s/step - loss: 0.4536 - ac
c: 0.7815
Epoch 73/100
60/60 [=====] - 119s 2s/step - loss: 0.4441 - ac
c: 0.7983
Epoch 74/100
60/60 [=====] - 120s 2s/step - loss: 0.4412 - ac
c: 0.7899
Epoch 75/100
60/60 [=====] - 125s 2s/step - loss: 0.4341 - ac
c: 0.8021
Epoch 76/100
60/60 [=====] - 121s 2s/step - loss: 0.4260 - ac
c: 0.8009
Epoch 77/100
60/60 [=====] - 113s 2s/step - loss: 0.4423 - ac
c: 0.7820
Epoch 78/100
60/60 [=====] - 120s 2s/step - loss: 0.4307 - ac
c: 0.7978
Epoch 79/100
60/60 [=====] - 123s 2s/step - loss: 0.4370 - ac
c: 0.8020
```

Epoch 80/100
60/60 [=====] - 112s 2s/step - loss: 0.4284 - acc: 0.7957
Epoch 81/100
60/60 [=====] - 119s 2s/step - loss: 0.4316 - acc: 0.7988
Epoch 82/100
60/60 [=====] - 114s 2s/step - loss: 0.4199 - acc: 0.8167
Epoch 83/100
60/60 [=====] - 112s 2s/step - loss: 0.4299 - acc: 0.8004
Epoch 84/100
60/60 [=====] - 112s 2s/step - loss: 0.4128 - acc: 0.8025
Epoch 85/100
60/60 [=====] - 113s 2s/step - loss: 0.4242 - acc: 0.8078
Epoch 86/100
60/60 [=====] - 112s 2s/step - loss: 0.4078 - acc: 0.8120
Epoch 87/100
60/60 [=====] - 111s 2s/step - loss: 0.4311 - acc: 0.7973
Epoch 88/100
60/60 [=====] - 111s 2s/step - loss: 0.4341 - acc: 0.7931
Epoch 89/100
60/60 [=====] - 110s 2s/step - loss: 0.4301 - acc: 0.7922
Epoch 90/100
60/60 [=====] - 112s 2s/step - loss: 0.4314 - acc: 0.8020
Epoch 91/100
60/60 [=====] - 111s 2s/step - loss: 0.4140 - acc: 0.8141
Epoch 92/100
60/60 [=====] - 112s 2s/step - loss: 0.4065 - acc: 0.8109
Epoch 93/100
60/60 [=====] - 109s 2s/step - loss: 0.4130 - acc: 0.8057
Epoch 94/100
60/60 [=====] - 109s 2s/step - loss: 0.4421 - acc: 0.7899
Epoch 95/100
60/60 [=====] - 111s 2s/step - loss: 0.4115 - acc: 0.8130
Epoch 96/100
60/60 [=====] - 111s 2s/step - loss: 0.4033 - acc: 0.8167
Epoch 97/100
60/60 [=====] - 107s 2s/step - loss: 0.4147 - acc: 0.8141
Epoch 98/100
60/60 [=====] - 109s 2s/step - loss: 0.4091 - acc: 0.8030
Epoch 99/100
60/60 [=====] - 116s 2s/step - loss: 0.4114 - acc: 0.8167
Epoch 100/100

60/60 [=====] - 111s 2s/step - loss: 0.4060 - ac
c: 0.8151

In [34]:

```
train_dir
```

Out[34]:

```
'C:\\kiran\\datasets\\cats_and_dogs_small\\train'
```

In [44]:

```
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=16,  
    # 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=16,  
    class_mode='binary')  
  
history = model.fit_generator(  
    train_generator,  
    steps_per_epoch=100,  
    epochs=60,  
    validation_data=validation_generator,  
    validation_steps=50)
```

```
Found 2000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
Epoch 1/60
100/100 [=====] - 77s 766ms/step - loss: 0.5326 -
acc: 0.7563 - val_loss: 0.6289 - val_acc: 0.7525
Epoch 2/60
100/100 [=====] - 98s 981ms/step - loss: 0.4868 -
acc: 0.7844 - val_loss: 0.5656 - val_acc: 0.7550
Epoch 3/60
100/100 [=====] - 101s 1s/step - loss: 0.4821 - a
cc: 0.7700 - val_loss: 0.4865 - val_acc: 0.7887
Epoch 4/60
100/100 [=====] - 108s 1s/step - loss: 0.4828 - a
cc: 0.7725 - val_loss: 0.6517 - val_acc: 0.6963
Epoch 5/60
100/100 [=====] - 126s 1s/step - loss: 0.4493 - a
cc: 0.8025 - val_loss: 0.5159 - val_acc: 0.7800
Epoch 6/60
100/100 [=====] - 126s 1s/step - loss: 0.4365 - a
cc: 0.8025 - val_loss: 0.4877 - val_acc: 0.7825
Epoch 7/60
100/100 [=====] - 133s 1s/step - loss: 0.4563 - a
cc: 0.7937 - val_loss: 0.4113 - val_acc: 0.8238
Epoch 8/60
100/100 [=====] - 126s 1s/step - loss: 0.4390 - a
cc: 0.7969 - val_loss: 0.4562 - val_acc: 0.8050
Epoch 9/60
100/100 [=====] - 113s 1s/step - loss: 0.4501 - a
cc: 0.7950 - val_loss: 0.4966 - val_acc: 0.7850
Epoch 10/60
100/100 [=====] - 115s 1s/step - loss: 0.4440 - a
cc: 0.7987 - val_loss: 0.5485 - val_acc: 0.7675
Epoch 11/60
100/100 [=====] - 111s 1s/step - loss: 0.4333 - a
cc: 0.8069 - val_loss: 0.4665 - val_acc: 0.7713
Epoch 12/60
100/100 [=====] - 112s 1s/step - loss: 0.4370 - a
cc: 0.7950 - val_loss: 0.4372 - val_acc: 0.8163
Epoch 13/60
100/100 [=====] - 115s 1s/step - loss: 0.4175 - a
cc: 0.8081 - val_loss: 0.4453 - val_acc: 0.8188
Epoch 14/60
100/100 [=====] - 114s 1s/step - loss: 0.4321 - a
cc: 0.8000 - val_loss: 0.4204 - val_acc: 0.8313
Epoch 15/60
100/100 [=====] - 116s 1s/step - loss: 0.4315 - a
cc: 0.8031 - val_loss: 0.5895 - val_acc: 0.7513
Epoch 16/60
100/100 [=====] - 115s 1s/step - loss: 0.4200 - a
cc: 0.8219 - val_loss: 0.4615 - val_acc: 0.8025
Epoch 17/60
100/100 [=====] - 130s 1s/step - loss: 0.4061 - a
cc: 0.8250 - val_loss: 0.4270 - val_acc: 0.8138
Epoch 18/60
100/100 [=====] - 114s 1s/step - loss: 0.4324 - a
cc: 0.8000 - val_loss: 0.4247 - val_acc: 0.8188
Epoch 19/60
100/100 [=====] - 113s 1s/step - loss: 0.4120 - a
cc: 0.8225 - val_loss: 0.5381 - val_acc: 0.7850
Epoch 20/60
100/100 [=====] - 114s 1s/step - loss: 0.4323 - a
```

```
cc: 0.8150 - val_loss: 0.4365 - val_acc: 0.8075
Epoch 21/60
100/100 [=====] - 113s 1s/step - loss: 0.4195 - a
cc: 0.8081 - val_loss: 0.4559 - val_acc: 0.8075
Epoch 22/60
100/100 [=====] - 111s 1s/step - loss: 0.4077 - a
cc: 0.8150 - val_loss: 0.4265 - val_acc: 0.8000
Epoch 23/60
100/100 [=====] - 115s 1s/step - loss: 0.4088 - a
cc: 0.8281 - val_loss: 0.4314 - val_acc: 0.8225
Epoch 24/60
100/100 [=====] - 111s 1s/step - loss: 0.4154 - a
cc: 0.8081 - val_loss: 0.4176 - val_acc: 0.8250
Epoch 25/60
100/100 [=====] - 113s 1s/step - loss: 0.4071 - a
cc: 0.8213 - val_loss: 0.4066 - val_acc: 0.8238
Epoch 26/60
100/100 [=====] - 113s 1s/step - loss: 0.3989 - a
cc: 0.8225 - val_loss: 0.4278 - val_acc: 0.8213
Epoch 27/60
100/100 [=====] - 113s 1s/step - loss: 0.3868 - a
cc: 0.8238 - val_loss: 0.4306 - val_acc: 0.7900
Epoch 28/60
100/100 [=====] - 115s 1s/step - loss: 0.3792 - a
cc: 0.8306 - val_loss: 0.4763 - val_acc: 0.7763
Epoch 29/60
100/100 [=====] - 116s 1s/step - loss: 0.4123 - a
cc: 0.8194 - val_loss: 0.4289 - val_acc: 0.8138
Epoch 30/60
100/100 [=====] - 116s 1s/step - loss: 0.3833 - a
cc: 0.8306 - val_loss: 0.3920 - val_acc: 0.8325
Epoch 31/60
100/100 [=====] - 128s 1s/step - loss: 0.3859 - a
cc: 0.8231 - val_loss: 0.3962 - val_acc: 0.8075
Epoch 32/60
100/100 [=====] - 126s 1s/step - loss: 0.3884 - a
cc: 0.8306 - val_loss: 0.4449 - val_acc: 0.8000
Epoch 33/60
100/100 [=====] - 127s 1s/step - loss: 0.3762 - a
cc: 0.8200 - val_loss: 0.5280 - val_acc: 0.7700
Epoch 34/60
100/100 [=====] - 126s 1s/step - loss: 0.3731 - a
cc: 0.8425 - val_loss: 0.4537 - val_acc: 0.8087
Epoch 35/60
100/100 [=====] - 116s 1s/step - loss: 0.3941 - a
cc: 0.8263 - val_loss: 0.4246 - val_acc: 0.8313
Epoch 36/60
100/100 [=====] - 114s 1s/step - loss: 0.3966 - a
cc: 0.8231 - val_loss: 0.4622 - val_acc: 0.8125
Epoch 37/60
100/100 [=====] - 116s 1s/step - loss: 0.3907 - a
cc: 0.8269 - val_loss: 0.5932 - val_acc: 0.7713
Epoch 38/60
100/100 [=====] - 115s 1s/step - loss: 0.3922 - a
cc: 0.8225 - val_loss: 0.4060 - val_acc: 0.8275
Epoch 39/60
100/100 [=====] - 116s 1s/step - loss: 0.3820 - a
cc: 0.8313 - val_loss: 0.4107 - val_acc: 0.8325
Epoch 40/60
100/100 [=====] - 111s 1s/step - loss: 0.3794 - a
cc: 0.8325 - val_loss: 0.5452 - val_acc: 0.7937
```

Epoch 41/60
100/100 [=====] - 114s 1s/step - loss: 0.3958 - acc: 0.8263 - val_loss: 0.4460 - val_acc: 0.8200

Epoch 42/60
100/100 [=====] - 112s 1s/step - loss: 0.3959 - acc: 0.8325 - val_loss: 0.4064 - val_acc: 0.8300

Epoch 43/60
100/100 [=====] - 118s 1s/step - loss: 0.3889 - acc: 0.8194 - val_loss: 0.4166 - val_acc: 0.8363

Epoch 44/60
100/100 [=====] - 118s 1s/step - loss: 0.3880 - acc: 0.8238 - val_loss: 0.4723 - val_acc: 0.8050

Epoch 45/60
100/100 [=====] - 119s 1s/step - loss: 0.3897 - acc: 0.8163 - val_loss: 0.5982 - val_acc: 0.7788

Epoch 46/60
100/100 [=====] - 117s 1s/step - loss: 0.3855 - acc: 0.8275 - val_loss: 0.4118 - val_acc: 0.8213

Epoch 47/60
100/100 [=====] - 113s 1s/step - loss: 0.3701 - acc: 0.8306 - val_loss: 0.3849 - val_acc: 0.8525

Epoch 48/60
100/100 [=====] - 117s 1s/step - loss: 0.3710 - acc: 0.8356 - val_loss: 0.5403 - val_acc: 0.7837

Epoch 49/60
100/100 [=====] - 116s 1s/step - loss: 0.3841 - acc: 0.8319 - val_loss: 0.6163 - val_acc: 0.7563

Epoch 50/60
100/100 [=====] - 115s 1s/step - loss: 0.3737 - acc: 0.8381 - val_loss: 0.4074 - val_acc: 0.8213

Epoch 51/60
100/100 [=====] - 116s 1s/step - loss: 0.3738 - acc: 0.8356 - val_loss: 0.4225 - val_acc: 0.8388

Epoch 52/60
100/100 [=====] - 116s 1s/step - loss: 0.3724 - acc: 0.8350 - val_loss: 0.6055 - val_acc: 0.7700

Epoch 53/60
100/100 [=====] - 109s 1s/step - loss: 0.3655 - acc: 0.8413 - val_loss: 0.4629 - val_acc: 0.8225

Epoch 54/60
100/100 [=====] - 111s 1s/step - loss: 0.3704 - acc: 0.8350 - val_loss: 0.4058 - val_acc: 0.8288

Epoch 55/60
100/100 [=====] - 108s 1s/step - loss: 0.3390 - acc: 0.8512 - val_loss: 0.4102 - val_acc: 0.8300

Epoch 56/60
100/100 [=====] - 106s 1s/step - loss: 0.3628 - acc: 0.8375 - val_loss: 0.4135 - val_acc: 0.8388

Epoch 57/60
100/100 [=====] - 108s 1s/step - loss: 0.3595 - acc: 0.8419 - val_loss: 0.4499 - val_acc: 0.8175

Epoch 58/60
100/100 [=====] - 107s 1s/step - loss: 0.3684 - acc: 0.8438 - val_loss: 0.4549 - val_acc: 0.8175

Epoch 59/60
100/100 [=====] - 107s 1s/step - loss: 0.3702 - acc: 0.8413 - val_loss: 0.3694 - val_acc: 0.8475

Epoch 60/60
100/100 [=====] - 108s 1s/step - loss: 0.3615 - acc: 0.8431 - val_loss: 0.4439 - val_acc: 0.8188

In [45]:

```
model.save('cats_and_dogs_small_2.h5')
```

In [46]:

```
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()
```

