In [ ]:

```
try:
  %tensorflow_version 2.x
except Exception:
  pass
import tensorflow as tf
tf.config.run_functions_eagerly(True)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D, Ir
from tensorflow.keras.preprocessing.image import ImageDataGenerator


import os
import numpy as np
import matplotlib.pyplot as plt
import scipy
```

In [ ]:

```
# Get project files

!unzip cats_and_dogs.zip

PATH = 'cats_and_dogs'

train_dir = os.path.join(PATH, 'train')
validation_dir = os.path.join(PATH, 'validation')
test_dir = os.path.join(PATH, 'test')

# Get number of files in each directory. The train and validation directories
# each have the subdirecories "dogs" and "cats".
total_train = sum([len(files) for r, d, files in os.walk(train_dir)])
total_val = sum([len(files) for r, d, files in os.walk(validation_dir)])
total_test = len(os.listdir(test_dir))

# Variables for pre-processing and training.
batch_size = 128
epochs = 15
IMG_HEIGHT = 150
IMG_WIDTH = 150
```

```
--2023-07-03 17:25:13--  https://cdn.freecodecamp.org/project-data/cats-and-dogs/cats
_and_dogs.zip
Resolving cdn.freecodecamp.org (cdn.freecodecamp.org)... 104.26.2.33, 104.26.3.33, 17
2.67.70.149
Connecting to cdn.freecodecamp.org (cdn.freecodecamp.org)|104.26.2.33|:443... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 70702765 (67M) [application/zip]
Saving to: 'cats_and_dogs.zip'

cats_and_dogs.zip   100%[===================>]  67,43M  2,07MB/s    in 33s

2023-07-03 17:25:47 (2,03 MB/s) - 'cats_and_dogs.zip' saved [70702765/70702765]

Archive:  cats_and_dogs.zip
   creating: cats_and_dogs/
  inflating: cats_and_dogs/.DS_Store
   creating: __MACOSX/
   creating: __MACOSX/cats_and_dogs/
  inflating: __MACOSX/cats_and_dogs/._.DS_Store
   creating: cats_and_dogs/test/
  inflating: cats_and_dogs/test/48.jpg
   creating: __MACOSX/cats_and_dogs/test/
  inflating: __MACOSX/cats_and_dogs/test/._48.jpg
```

```
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2270.jpg
 inflating: cats_and_dogs/validation/cats/cat.2258.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2258.jpg
 inflating: cats_and_dogs/validation/cats/cat.2476.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2476.jpg
 inflating: cats_and_dogs/validation/cats/cat.2310.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2310.jpg
 inflating: cats_and_dogs/validation/cats/cat.2304.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2304.jpg
 inflating: cats_and_dogs/validation/cats/cat.2462.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2462.jpg
 inflating: cats_and_dogs/validation/cats/cat.2338.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2338.jpg
 inflating: cats_and_dogs/validation/cats/cat.2489.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2489.jpg
 inflating: cats_and_dogs/validation/cats/cat.2112.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2112.jpg
 inflating: cats_and_dogs/validation/cats/cat.2106.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2106.jpg
 inflating: cats_and_dogs/validation/cats/cat.2107.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2107.jpg
 inflating: cats_and_dogs/validation/cats/cat.2113.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2113.jpg
 inflating: cats_and_dogs/validation/cats/cat.2488.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2488.jpg
 inflating: cats_and_dogs/validation/cats/cat.2339.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2339.jpg
 inflating: cats_and_dogs/validation/cats/cat.2305.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2305.jpg
 inflating: cats_and_dogs/validation/cats/cat.2463.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2463.jpg
 inflating: cats_and_dogs/validation/cats/cat.2477.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2477.jpg
 inflating: cats_and_dogs/validation/cats/cat.2311.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2311.jpg
 inflating: cats_and_dogs/validation/cats/cat.2259.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2259.jpg
 inflating: cats_and_dogs/validation/cats/cat.2271.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2271.jpg
 inflating: cats_and_dogs/validation/cats/cat.2265.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2265.jpg
 inflating: cats_and_dogs/validation/cats/cat.2098.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2098.jpg
 inflating: cats_and_dogs/validation/cats/cat.2073.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2073.jpg
 inflating: cats_and_dogs/validation/cats/cat.2067.jpg
 inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2067.jpg
 inflating: __MACOSX/cats_and_dogs/validation/._cats
 inflating: cats_and_dogs/validation/.DS_Store
 inflating: __MACOSX/cats_and_dogs/validation/._.DS_Store
 inflating: __MACOSX/cats_and_dogs/._validation
 inflating: __MACOSX/._cats_and_dogs
```

In [ ]:

```
# 3

# Create image generators for each dataset

rescale=1/255

train_image_generator = ImageDataGenerator(rescale=rescale)
validation_image_generator = ImageDataGenerator(rescale=rescale)
test_image_generator = ImageDataGenerator(rescale=rescale)

train_data_gen = train_image_generator.flow_from_directory(
    batch_size=batch_size,
    directory=train_dir,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    class_mode='binary')
```

```
val_data_gen = validation_image_generator.flow_from_directory(
    batch_size=batch_size,
    directory=validation_dir,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    class_mode='binary')

test_data_gen = test_image_generator.flow_from_directory(
    batch_size=batch_size,
    classes=["."], # this is the trick bit
    directory=test_dir,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    shuffle=False,
    class_mode="binary")
```

```
Found 2000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
Found 50 images belonging to 1 classes.
```

In [ ]:

```
# 4
def plotImages(images_arr, probabilities = False):
    fig, axes = plt.subplots(len(images_arr), 1, figsize=(5,len(images_arr) * 3))
    if probabilities is False:
      for img, ax in zip( images_arr, axes):
          ax.imshow(img)
          ax.axis('off')
    else:
      for img, probability, ax in zip( images_arr, probabilities, axes):
          ax.imshow(img)
          ax.axis('off')
          if probability > 0.5:
              ax.set_title("%.2f" % (probability*100) + "% dog")
          else:
              ax.set_title("%.2f" % ((1-probability)*100) + "% cat")
    plt.show()

sample_training_images, _ = next(train_data_gen)
plotImages(sample_training_images[:5])
```

In [ ]:

```
# 5
# Recreate the train_image_generator using ImageDataGenerator with random transformat

train_image_generator = ImageDataGenerator(
    rescale=rescale,
    horizontal_flip=True,
    rotation_range=20,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    fill_mode="nearest"
)
```

In [ ]:

```
# 6

train_data_gen = train_image_generator.flow_from_directory(
    batch_size=batch_size,
    directory=train_dir,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    class_mode='binary')
```

```
augmented_images = [train_data_gen[0][0][0] for i in range(5)]

plotImages(augmented_images)
```

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

In [ ]:

```
# 7
# Create the model
model = Sequential([
    Input(shape=(IMG_HEIGHT, IMG_WIDTH, 3)),
    Conv2D(16, 3, padding='same', activation='relu'),
    MaxPooling2D(),
    Conv2D(32, 3, padding='same', activation='relu'),
    MaxPooling2D(),
    Conv2D(64, 3, padding='same', activation='relu'),
    MaxPooling2D(),
    Dropout(0.2),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(2)
])

model.compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['accuracy'],
)

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 150, 150, 16)      448

 max_pooling2d (MaxPooling2D  (None, 75, 75, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 75, 75, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 37, 37, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 37, 37, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 18, 18, 64)       0
 2D)

 dropout (Dropout)           (None, 18, 18, 64)        0

 flatten (Flatten)           (None, 20736)             0

 dense (Dense)               (None, 128)               2654336

 dense_1 (Dense)             (None, 2)                 258

=================================================================
Total params: 2,678,178
Trainable params: 2,678,178
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

```
from math import ceil
```

```
steps_per_epoch=ceil(len(train_data_gen) / batch_size)
steps_per_epoch
```

Out[ ]:
```
1
```

In [ ]:
```
validation_steps = ceil(len(val_data_gen) / batch_size)
validation_steps
```

Out[ ]:
```
1
```

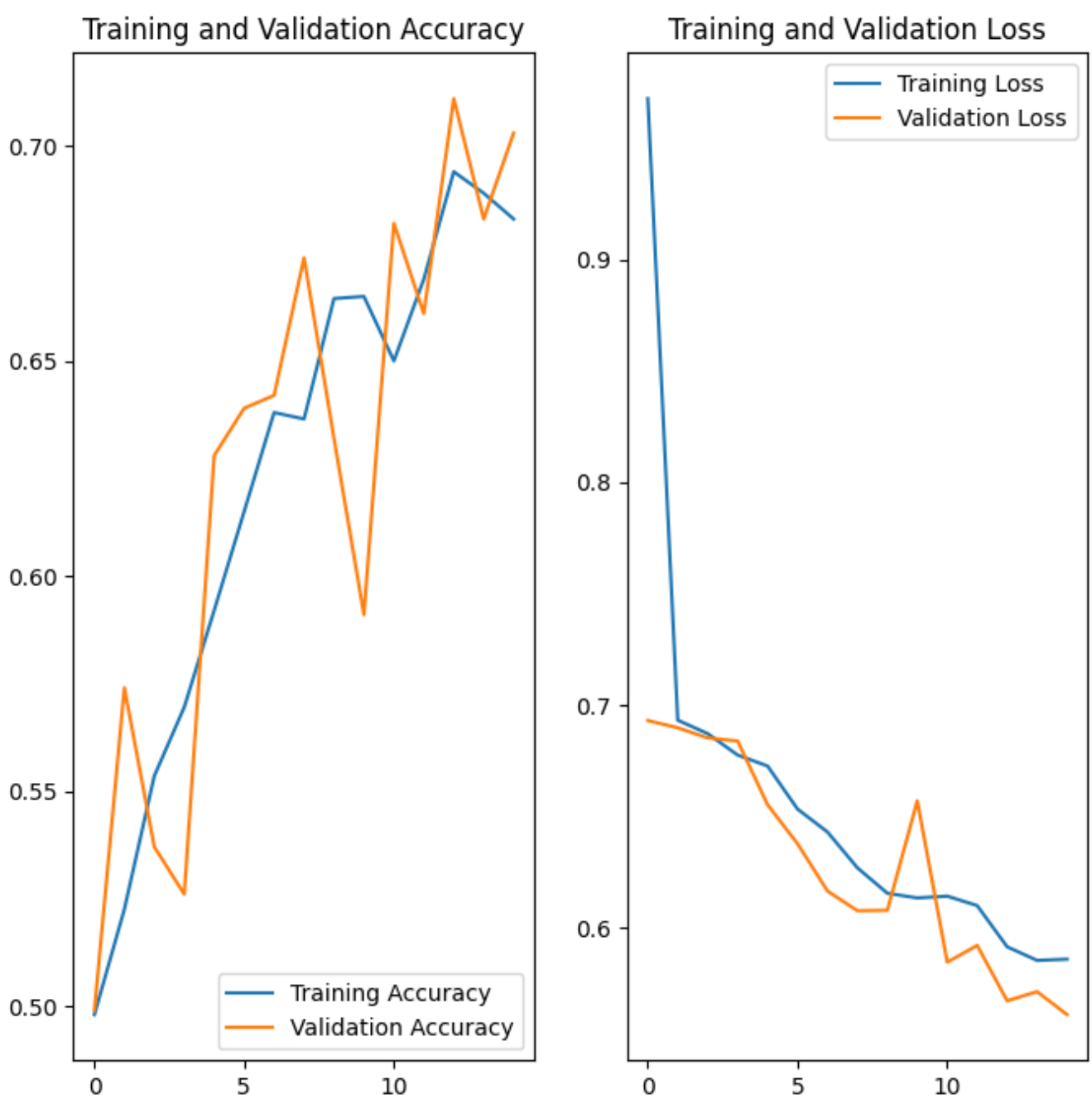
In [ ]:
```
# 8
# Train the model
history = model.fit(
    train_data_gen,
    epochs=epochs,
    validation_data=val_data_gen,
)
```

```
/usr/local/lib/python3.9/site-packages/tensorflow/python/data/ops/structured_functio
n.py:254: UserWarning: Even though the `tf.config.experimental_run_functions_eagerly`
option is set, this option does not apply to tf.data functions. To force eager execut
ion of tf.data functions, please use `tf.data.experimental.enable_debug_mode()`.
  warnings.warn(
2023-07-03 17:31:54.006160: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_0' with dtype int32
         [[{{node Placeholder/_0}}]]
Epoch 1/15
16/16 [==============================] - ETA: 0s - loss: 0.9722 - accuracy: 0.4980
2023-07-03 17:32:54.473686: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_0' with dtype int32
         [[{{node Placeholder/_0}}]]
16/16 [==============================] - 71s 4s/step - loss: 0.9722 - accuracy: 0.498
0 - val_loss: 0.6932 - val_accuracy: 0.4990
Epoch 2/15
16/16 [==============================] - 67s 4s/step - loss: 0.6933 - accuracy: 0.522
5 - val_loss: 0.6899 - val_accuracy: 0.5740
Epoch 3/15
16/16 [==============================] - 62s 4s/step - loss: 0.6873 - accuracy: 0.553
5 - val_loss: 0.6854 - val_accuracy: 0.5370
Epoch 4/15
16/16 [==============================] - 62s 4s/step - loss: 0.6777 - accuracy: 0.569
5 - val_loss: 0.6840 - val_accuracy: 0.5260
Epoch 5/15
16/16 [==============================] - 61s 4s/step - loss: 0.6728 - accuracy: 0.592
0 - val_loss: 0.6554 - val_accuracy: 0.6280
Epoch 6/15
16/16 [==============================] - 62s 4s/step - loss: 0.6535 - accuracy: 0.615
0 - val_loss: 0.6379 - val_accuracy: 0.6390
Epoch 7/15
16/16 [==============================] - 64s 4s/step - loss: 0.6432 - accuracy: 0.638
0 - val_loss: 0.6167 - val_accuracy: 0.6420
Epoch 8/15
16/16 [==============================] - 63s 4s/step - loss: 0.6271 - accuracy: 0.636
5 - val_loss: 0.6079 - val_accuracy: 0.6740
Epoch 9/15
16/16 [==============================] - 65s 4s/step - loss: 0.6157 - accuracy: 0.664
5 - val_loss: 0.6081 - val_accuracy: 0.6320
Epoch 10/15
16/16 [==============================] - 64s 4s/step - loss: 0.6136 - accuracy: 0.665
0 - val_loss: 0.6572 - val_accuracy: 0.5910
Epoch 11/15
```

```
16/16 [==============================] - 64s 4s/step - loss: 0.6144 - accuracy: 0.650
0 - val_loss: 0.5848 - val_accuracy: 0.6820
Epoch 12/15
16/16 [==============================] - 60s 4s/step - loss: 0.6102 - accuracy: 0.669
0 - val_loss: 0.5923 - val_accuracy: 0.6610
Epoch 13/15
16/16 [==============================] - 62s 4s/step - loss: 0.5917 - accuracy: 0.694
0 - val_loss: 0.5675 - val_accuracy: 0.7110
Epoch 14/15
16/16 [==============================] - 62s 4s/step - loss: 0.5856 - accuracy: 0.689
0 - val_loss: 0.5716 - val_accuracy: 0.6830
Epoch 15/15
16/16 [==============================] - 60s 4s/step - loss: 0.5861 - accuracy: 0.683
0 - val_loss: 0.5613 - val_accuracy: 0.7030
```

In [ ]:

```python
# 9
# Visualize the accuracy and loss

acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

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

epochs_range = range(epochs)

plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



In [ ]:

```
# Predict the probabilities of test images

predictions = model.predict(test_data_gen)
probabilities = [1 if a[0]<a[1] else 0 for a in predictions]
```

```
2023-07-03 17:48:49.687103: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_0' with dtype int32
         [[{{node Placeholder/_0}}]]
1/1 [==============================] - 1s 521ms/step
```

In [ ]:

```
print(len(probabilities))
```

```
50
```

In [ ]:

```
# 11
answers =  [1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,
            1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0,
            1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1,
            1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1,
            0, 0, 0, 0, 0, 0]

correct = 0
```

```
for probability, answer in zip(probabilities, answers):
  if round(probability) == answer:
    correct +=1

percentage_identified = (correct / len(answers))

passed_challenge = percentage_identified > 0.63

print( f"Your model correctly identified {round(percentage_identified*100, 2)}% "
       f"of the images of cats and dogs.")

if passed_challenge:
  print("You passed the challenge!")
else:
  print("You haven't passed yet. Your model should identify at least 63% "
        "of the images. Keep trying. You will get it!")
```

```
Your model correctly identified 76.0% of the images of cats and dogs.
```