```
try:
  # This command only in Colab.
  %tensorflow_version 2.x
except Exception:
  pass
import tensorflow as tf

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

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

```
    Colab only includes TensorFlow 2.x; %tensorflow_version has no effect.
```

```
# Get project files
!wget https://cdn.freecodecamp.org/project-data/cats-and-dogs/cats_and_dogs.zip

!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
```

```
      inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2072.jpg
      inflating: cats_and_dogs/validation/cats/cat.2099.jpg
      inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2099.jpg
      inflating: cats_and_dogs/validation/cats/cat.2264.jpg
      inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2264.jpg
      inflating: cats_and_dogs/validation/cats/cat.2270.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
```

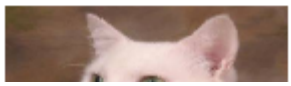
```
  inflating: cats_and_dogs/validation/cats/cat.2462.jpg
  inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2462.jpg
  inflating: cats_and_dogs/validation/cats/cat.2338.jpg
  inflating: __MACOSX/cats_and_dogs/validation/cats/._cat.2338.jpg
  inflating: cats_and_dogs/validation/cats/cat.2489.jpg
  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
```

```
# 3
train_image_generator = ImageDataGenerator(rescale=1./255)
validation_image_generator = ImageDataGenerator(rescale=1./255)
test_image_generator = ImageDataGenerator(rescale=1./255)

train_data_gen = train_image_generator.flow_from_directory(directory=train_dir,
                                                           batch_size = batch_size
                                                           target_size =(IMG_HEIGH
                                                           class_mode='binary')
val_data_gen = validation_image_generator.flow_from_directory(directory=validation_
                                                              batch_size=batch_siz
                                                              target_size=(IMG_HEI
                                                              class_mode='binary')
```

```
test_data_gen = test_image_generator.flow_from_directory(directory=PATH,
                                                         classes=['test'],
                                                         target_size=(IMG_HEIGHT, I
                                                         batch_size=batch_size,
                                                         shuffle=False)
```

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

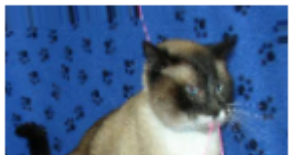
```
# 4
def plotImages(images_arr, probabilities = False):
    fig, axes = plt.subplots(len(images_arr), 1, figsize=(5,len(images_arr) * 3))
    if probabilities is False:
      for img, ax in zip( images_arr, axes):
          ax.imshow(img)
          ax.axis('off')
    else:
      for img, probability, 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])
```



```
# 5
train_image_generator = ImageDataGenerator(rescale=1./255,
                                           shear_range=0.2,
                                           zoom_range=0.2,
                                           rotation_range=25,
                                           horizontal_flip=True,
                                           vertical_flip=True,
                                           validation_split=.2)
```

```
# 6
train_data_gen = train_image_generator.flow_from_directory(batch_size=batch_size,
                                                           directory=train_dir,
                                                           target_size=(IMG_HEIGHT, IMG_
                                                           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.
```



```
# 7
model = Sequential()

model.add(Conv2D(32, (3,3), activation='relu', input_shape=(IMG_HEIGHT, IMG_WIDTH,1
model.add(MaxPooling2D((2,2)))

model.add(Conv2D(64, (3,3), activation='relu'))
model.add(MaxPooling2D((2,2)))

model.add(Conv2D(128, (3,3), activation='relu'))
model.add(MaxPooling2D((2,2)))

model.add(Flatten())
model.add(Dense(64))
model.add(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, 148, 148, 32)      896

 max_pooling2d (MaxPooling2D  (None, 74, 74, 32)       0
 )

 conv2d_1 (Conv2D)            (None, 72, 72, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 64)       0
 2D)

 conv2d_2 (Conv2D)            (None, 34, 34, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 128)      0
 2D)

 flatten (Flatten)            (None, 36992)             0

 dense (Dense)                (None, 64)                2367552

 dense_1 (Dense)              (None, 2)                 130

=================================================================
Total params: 2,460,930
Trainable params: 2,460,930
Non-trainable params: 0
_________________________________________________________________
```

```
# 8
history = model.fit(x=train_data_gen,
                    steps_per_epoch=total_train // batch_size,
                    epochs=epochs,
                    validation_data=val_data_gen,
                    validation_steps=total_val // batch_size)
```

```
Epoch 1/15
15/15 [==============================] - 111s 7s/step - loss: 1.2076 - accura
Epoch 2/15
15/15 [==============================] - 104s 7s/step - loss: 0.6873 - accura
Epoch 3/15
15/15 [==============================] - 106s 7s/step - loss: 0.6779 - accura
Epoch 4/15
15/15 [==============================] - 104s 7s/step - loss: 0.6725 - accura
Epoch 5/15
15/15 [==============================] - 103s 7s/step - loss: 0.6571 - accura
Epoch 6/15
15/15 [==============================] - 106s 7s/step - loss: 0.6336 - accura
Epoch 7/15
15/15 [==============================] - 106s 7s/step - loss: 0.6332 - accura
Epoch 8/15
```

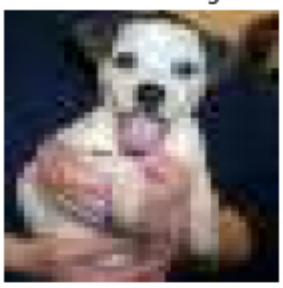
```
15/15 [==============================] - 102s 7s/step - loss: 0.6232 - accura
Epoch 9/15
15/15 [==============================] - 103s 7s/step - loss: 0.6123 - accura
Epoch 10/15
15/15 [==============================] - 106s 7s/step - loss: 0.5974 - accura
Epoch 11/15
15/15 [==============================] - 103s 7s/step - loss: 0.5913 - accura
Epoch 12/15
15/15 [==============================] - 108s 7s/step - loss: 0.5905 - accura
Epoch 13/15
15/15 [==============================] - 111s 7s/step - loss: 0.5817 - accura
Epoch 14/15
15/15 [==============================] - 107s 7s/step - loss: 0.5698 - accura
Epoch 15/15
15/15 [==============================] - 103s 7s/step - loss: 0.5729 - accura
```

```
# 9
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()
```


```
# 10
test_images, _ = next(test_data_gen)
probabilities = np.argmax(model.predict(test_data_gen), axis=-1)
plotImages(test_images, probabilities=probabilities)
```

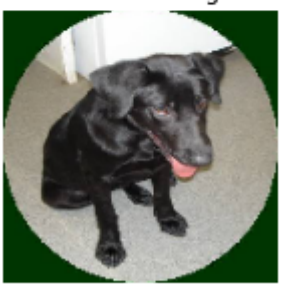
100.00% cat

100.00% dog



100.00% cat

100.00% dog



100.00% dog

100.00% cat