```
#Load tensorflow
import tensorflow as tf
import glob

#create excel data
import pandas as pd1
import pandas as pd2
df_train = pd1.DataFrame(columns=['class','dir','image_file'])
df_test = pd2.DataFrame(columns=['class','dir','image_file'])

train_cats_dogs=list()
train_cats_dogs_class=list()
test_cats_dogs=list()
test_cats_dogs_class=list()

#Parse .jpg image files located in /data/*/*/*jpg and make train, test csv.

for entry in glob.glob('./data/*/*/*jpg'):
    cats_dogs = entry
    if "/train" in cats_dogs:
        train_cats_dogs.append(cats_dogs)
        cats_dogs_dir = cats_dogs[0:cats_dogs.rfind('/')]
        cats_dogs_class = cats_dogs_dir[cats_dogs_dir.rfind('/')+1:]
        df_train.loc[df_train.shape[0]] = [cats_dogs_class, cats_dogs_dir, cats_dogs]


    if "/test" in cats_dogs:
        test_cats_dogs.append(cats_dogs)
        cats_dogs_dir = cats_dogs[0:cats_dogs.rfind('/')]
        cats_dogs_class = cats_dogs_dir[cats_dogs_dir.rfind('/')+1:]
        df_test.loc[df_test.shape[0]] = [cats_dogs_class, cats_dogs_dir, cats_dogs]


print('Total images: ', df_train.shape[0])
df_train.head()

print('Total images: ', df_test.shape[0])
df_test.head()

Total images:  40
Total images:  20
class   dir image_file
0   cats    ./data/test/cats    ./data/test/cats/110.jpg
1   cats    ./data/test/cats    ./data/test/cats/102.jpg
2   cats    ./data/test/cats    ./data/test/cats/107.jpg
3   cats    ./data/test/cats    ./data/test/cats/105.jpg
4   cats    ./data/test/cats    ./data/test/cats/109.jpg

#images by class
print(df_train.groupby(['class']).size())
print(df_test.groupby(['class']).size())

#create .csv files
df_train.to_csv('data/train.csv',index=False)
df_test.to_csv('data/test.csv',index=False)

class
cats    20
dogs    20
dtype: int64
class
cats    10
dogs    10
dtype: int64

#Get class names
class_names = df_train['class'].unique().tolist()
print('Animal classes: ', class_names)
Animal classes:  ['cats', 'dogs']
```

```
#Define some parameters
img_size = 60
img_depth = 3


#Build batch generator
import numpy as np
def batch_generator(df, batchsize=32, train_mode=True):

    img_generator= tf.keras.preprocessing.image.ImageDataGenerator(rotation_range=20,
                                                                    width_shift_range=0.2,

                                                                    height_shift_range=0.2
                                                                    ,
                                                                    horizontal_flip=True,

                                                                    featurewise_center=Tru
                                                                    e)


    while True:

                                img_class = df.loc[batch_nums[i]]['class']
        #Generate random numbers to pick images from dataset
        batch_nums = np.random.randint(0,df.shape[0],batchsize)

        #Initialize batch images array
        batch_images = np.zeros((batchsize,img_size, img_size,img_depth))

        #Initiate batch label array
        batch_labels = np.zeros((batchsize, len(class_names)))

        for i in range(batchsize):

            #Load image
            animal_image =  tf.keras.preprocessing.image.load_img(

                                                                df.loc[batch_nums[i]]['
                                                                image_file'],

                                                                target_size=(img_size,
                                                                img_size))
            #Convert to array
            animal_image = tf.keras.preprocessing.image.img_to_array(animal_image)

            if(train_mode):
                #Apply transform
                animal_image =  img_generator.random_transform(animal_image)


            #Get the class
            img_class = df.loc[batch_nums[i]]['class']
            #Conver class to number
            img_class = class_names.index(img_class)
            #Convert class to one hot encoding
            img_class = tf.keras.utils.to_categorical(img_class,
            num_classes=len(class_names))

            #Update batch images and class arrays
            batch_images[i] = animal_image
            batch_labels[i] = img_class
            #print("I value:", i)

        yield batch_images, batch_labels


#Build CNN Model
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.BatchNormalization(input_shape=(img_size,img_size,3,)))
```

```
model.add(tf.keras.layers.Conv2D(32, kernel_size=(5,5), strides=2,
activation=tf.keras.layers.LeakyReLU(alpha=0.05)))
model.add(tf.keras.layers.MaxPool2D(pool_size=(2,2)))
model.add(tf.keras.layers.Conv2D(64, kernel_size=(5,5), strides=2,
activation=tf.keras.layers.LeakyReLU(alpha=0.05)))
model.add(tf.keras.layers.MaxPool2D(pool_size=(2,2)))
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(200, activation=tf.keras.layers.LeakyReLU(alpha=0.03)))
model.add(tf.keras.layers.Dropout(0.4))
#model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(64, activation=tf.keras.layers.LeakyReLU(alpha=0.03)))
#Output layer
model.add(tf.keras.layers.Dense(len(class_names), activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

#Train the model
#Create train and test generator
batchsize = 64
train_generator = batch_generator(df_train, batchsize=batchsize) #batchsize can be
changed
test_generator = batch_generator(df_test, batchsize=batchsize, train_mode=False)

model.fit_generator(train_generator,
                          epochs=100,
                          steps_per_epoch=64,
                          validation_data=test_generator,
                          validation_steps=32,
                          shuffle=True)
WARNING:tensorflow:sample_weight modes were coerced from
  ...
    to
  ['...']
WARNING:tensorflow:sample_weight modes were coerced from
  ...
    to
  ['...']
Train for 64 steps, validate for 32 steps
Epoch 1/100
64/64 [==============================] - 25s 385ms/step - loss: 0.6245 - accuracy:
0.6382 - val_loss: 0.9095 - val_accuracy: 0.5562
Epoch 2/100
64/64 [==============================] - 23s 360ms/step - loss: 0.3773 - accuracy:
0.8265 - val_loss: 0.8330 - val_accuracy: 0.7283
Epoch 3/100
64/64 [==============================] - 24s 368ms/step - loss: 0.2384 - accuracy:
0.9003 - val_loss: 0.6090 - val_accuracy: 0.8032
Epoch 4/100
64/64 [==============================] - 23s 362ms/step - loss: 0.1225 - accuracy:
0.9539 - val_loss: 0.9444 - val_accuracy: 0.5913
Epoch 5/100
64/64 [==============================] - 23s 362ms/step - loss: 0.1008 - accuracy:
0.9634 - val_loss: 1.3734 - val_accuracy: 0.6460
Epoch 6/100
64/64 [==============================] - 23s 364ms/step - loss: 0.0693 - accuracy:
0.9767 - val_loss: 1.0670 - val_accuracy: 0.6333
Epoch 7/100
64/64 [==============================] - 23s 361ms/step - loss: 0.0297 - accuracy:
0.9902 - val_loss: 1.3865 - val_accuracy: 0.6958
Epoch 8/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0460 - accuracy:
0.9833 - val_loss: 1.3593 - val_accuracy: 0.5620
Epoch 9/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0343 - accuracy:
0.9905 - val_loss: 1.6068 - val_accuracy: 0.7314
Epoch 10/100
64/64 [==============================] - 23s 367ms/step - loss: 0.0390 - accuracy:
0.9851 - val_loss: 1.9976 - val_accuracy: 0.5732
Epoch 11/100
64/64 [==============================] - 23s 362ms/step - loss: 0.0199 - accuracy:
```

```
0.9933 - val_loss: 2.4647 - val_accuracy: 0.6582
Epoch 12/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0127 - accuracy:
0.9957 - val_loss: 2.6516 - val_accuracy: 0.5693
Epoch 13/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0317 - accuracy:
0.9894 - val_loss: 2.3742 - val_accuracy: 0.5679
Epoch 14/100
64/64 [==============================] - 23s 365ms/step - loss: 0.0153 - accuracy:
0.9949 - val_loss: 2.9684 - val_accuracy: 0.5493
Epoch 15/100
64/64 [==============================] - 23s 360ms/step - loss: 0.0211 - accuracy:
0.9935 - val_loss: 2.9727 - val_accuracy: 0.5120
Epoch 16/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0129 - accuracy:
0.9952 - val_loss: 2.7030 - val_accuracy: 0.5894
Epoch 17/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0071 - accuracy:
0.9976 - val_loss: 2.9585 - val_accuracy: 0.5342
Epoch 18/100
64/64 [==============================] - 23s 361ms/step - loss: 0.0135 - accuracy:
0.9954 - val_loss: 3.0759 - val_accuracy: 0.5952
Epoch 19/100
64/64 [==============================] - 23s 364ms/step - loss: 0.0278 - accuracy:
0.9891 - val_loss: 2.8800 - val_accuracy: 0.6089
Epoch 20/100
64/64 [==============================] - 23s 364ms/step - loss: 0.0155 - accuracy:
0.9952 - val_loss: 3.8784 - val_accuracy: 0.6567
Epoch 21/100
64/64 [==============================] - 23s 362ms/step - loss: 0.0132 - accuracy:
0.9967 - val_loss: 3.3587 - val_accuracy: 0.6445
Epoch 22/100
64/64 [==============================] - 23s 364ms/step - loss: 0.0131 - accuracy:
0.9952 - val_loss: 2.7973 - val_accuracy: 0.6357
Epoch 23/100
64/64 [==============================] - 23s 361ms/step - loss: 0.0280 - accuracy:
0.9913 - val_loss: 3.2062 - val_accuracy: 0.6514
Epoch 24/100
64/64 [==============================] - 23s 367ms/step - loss: 0.0331 - accuracy:
0.9884 - val_loss: 2.7728 - val_accuracy: 0.5386
Epoch 25/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0115 - accuracy:
0.9965 - val_loss: 2.4382 - val_accuracy: 0.5654
Epoch 26/100
64/64 [==============================] - 24s 370ms/step - loss: 0.0039 - accuracy:
0.9991 - val_loss: 4.0387 - val_accuracy: 0.5820
Epoch 27/100
64/64 [==============================] - 23s 359ms/step - loss: 0.0112 - accuracy:
0.9980 - val_loss: 3.4405 - val_accuracy: 0.6992
Epoch 28/100
64/64 [==============================] - 24s 368ms/step - loss: 0.0045 - accuracy:
0.9987 - val_loss: 3.7080 - val_accuracy: 0.6445
Epoch 29/100
64/64 [==============================] - 23s 367ms/step - loss: 0.0016 - accuracy:
0.9995 - val_loss: 3.7287 - val_accuracy: 0.6118
Epoch 30/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0019 - accuracy:
0.9996 - val_loss: 3.3922 - val_accuracy: 0.6035
Epoch 31/100
64/64 [==============================] - 24s 369ms/step - loss: 0.0043 - accuracy:
0.9989 - val_loss: 3.9576 - val_accuracy: 0.5879
Epoch 32/100
64/64 [==============================] - 23s 364ms/step - loss: 0.0016 - accuracy:
0.9999 - val_loss: 4.0844 - val_accuracy: 0.5967
Epoch 33/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0126 - accuracy:
0.9963 - val_loss: 2.5933 - val_accuracy: 0.6311
Epoch 34/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0097 - accuracy:
```

```
0.9967 - val_loss: 3.1865 - val_accuracy: 0.6726
Epoch 35/100
64/64 [==============================] - 23s 367ms/step - loss: 0.0043 - accuracy:
0.9984 - val_loss: 4.2086 - val_accuracy: 0.6123
Epoch 36/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0085 - accuracy:
0.9976 - val_loss: 3.0882 - val_accuracy: 0.6475
Epoch 37/100
64/64 [==============================] - 23s 365ms/step - loss: 0.0300 - accuracy:
0.9907 - val_loss: 2.9431 - val_accuracy: 0.5967
Epoch 38/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0190 - accuracy:
0.9945 - val_loss: 2.4969 - val_accuracy: 0.7051
Epoch 39/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0092 - accuracy:
0.9974 - val_loss: 3.6264 - val_accuracy: 0.7017
Epoch 40/100
64/64 [==============================] - 24s 368ms/step - loss: 0.0036 - accuracy:
0.9987 - val_loss: 4.6557 - val_accuracy: 0.5464
Epoch 41/100
64/64 [==============================] - 23s 365ms/step - loss: 0.0016 - accuracy:
0.9996 - val_loss: 4.2020 - val_accuracy: 0.5972
Epoch 42/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0023 - accuracy:
0.9989 - val_loss: 4.1630 - val_accuracy: 0.5610
Epoch 43/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0038 - accuracy:
0.9988 - val_loss: 3.6876 - val_accuracy: 0.5991
Epoch 44/100
64/64 [==============================] - 23s 362ms/step - loss: 0.0016 - accuracy:
0.9994 - val_loss: 4.2227 - val_accuracy: 0.6484
Epoch 45/100
64/64 [==============================] - 23s 367ms/step - loss: 0.0057 - accuracy:
0.9971 - val_loss: 3.3744 - val_accuracy: 0.5679
Epoch 46/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0100 - accuracy:
0.9962 - val_loss: 2.9655 - val_accuracy: 0.7305
Epoch 47/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0127 - accuracy:
0.9961 - val_loss: 3.9505 - val_accuracy: 0.5498
Epoch 48/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0125 - accuracy:
0.9958 - val_loss: 5.0354 - val_accuracy: 0.5562
Epoch 49/100
64/64 [==============================] - 24s 372ms/step - loss: 0.0185 - accuracy:
0.9943 - val_loss: 4.9747 - val_accuracy: 0.4541
Epoch 50/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0241 - accuracy:
0.9934 - val_loss: 3.1141 - val_accuracy: 0.6294
Epoch 51/100
64/64 [==============================] - 23s 364ms/step - loss: 0.0023 - accuracy:
0.9995 - val_loss: 3.1312 - val_accuracy: 0.5562
Epoch 52/100
64/64 [==============================] - 24s 370ms/step - loss: 4.2922e-04 - accuracy:
1.0000 - val_loss: 3.5130 - val_accuracy: 0.5483
Epoch 53/100
64/64 [==============================] - 23s 364ms/step - loss: 3.9033e-04 - accuracy:
1.0000 - val_loss: 4.1483 - val_accuracy: 0.5898
Epoch 54/100
64/64 [==============================] - 24s 368ms/step - loss: 2.4888e-04 - accuracy:
1.0000 - val_loss: 4.5619 - val_accuracy: 0.5522
Epoch 55/100
64/64 [==============================] - 24s 368ms/step - loss: 5.8750e-04 - accuracy:
0.9998 - val_loss: 4.1330 - val_accuracy: 0.6978
Epoch 56/100
64/64 [==============================] - 23s 367ms/step - loss: 0.0202 - accuracy:
0.9940 - val_loss: 3.9637 - val_accuracy: 0.5776
Epoch 57/100
64/64 [==============================] - 24s 369ms/step - loss: 0.0648 - accuracy:
```

```
0.9819 - val_loss: 2.7062 - val_accuracy: 0.7329
Epoch 58/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0038 - accuracy:
0.9988 - val_loss: 2.9132 - val_accuracy: 0.6218
Epoch 59/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0038 - accuracy:
0.9987 - val_loss: 2.8939 - val_accuracy: 0.6353
Epoch 60/100
64/64 [==============================] - 23s 367ms/step - loss: 9.3284e-04 - accuracy:
1.0000 - val_loss: 3.6556 - val_accuracy: 0.5864
Epoch 61/100
64/64 [==============================] - 23s 367ms/step - loss: 8.0939e-04 - accuracy:
0.9998 - val_loss: 3.2075 - val_accuracy: 0.6499
Epoch 62/100
64/64 [==============================] - 23s 367ms/step - loss: 0.0015 - accuracy:
0.9993 - val_loss: 3.2318 - val_accuracy: 0.6562
Epoch 63/100
64/64 [==============================] - 24s 367ms/step - loss: 0.0037 - accuracy:
0.9993 - val_loss: 3.8935 - val_accuracy: 0.6069
Epoch 64/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0106 - accuracy:
0.9968 - val_loss: 2.8898 - val_accuracy: 0.6416
Epoch 65/100
64/64 [==============================] - 24s 367ms/step - loss: 0.0016 - accuracy:
0.9991 - val_loss: 2.8350 - val_accuracy: 0.6426
Epoch 66/100
64/64 [==============================] - 23s 365ms/step - loss: 0.0066 - accuracy:
0.9983 - val_loss: 3.2272 - val_accuracy: 0.5977
Epoch 67/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0013 - accuracy:
0.9995 - val_loss: 3.9669 - val_accuracy: 0.5967
Epoch 68/100
64/64 [==============================] - 23s 365ms/step - loss: 0.0012 - accuracy:
0.9995 - val_loss: 3.8911 - val_accuracy: 0.5552
Epoch 69/100
64/64 [==============================] - 23s 363ms/step - loss: 5.7485e-04 - accuracy:
1.0000 - val_loss: 3.8883 - val_accuracy: 0.6045
Epoch 70/100
64/64 [==============================] - 23s 365ms/step - loss: 3.3674e-04 - accuracy:
1.0000 - val_loss: 4.0434 - val_accuracy: 0.5649
Epoch 71/100
64/64 [==============================] - 23s 366ms/step - loss: 1.2648e-04 - accuracy:
1.0000 - val_loss: 3.8027 - val_accuracy: 0.5986
Epoch 72/100
64/64 [==============================] - 23s 361ms/step - loss: 4.0793e-04 - accuracy:
1.0000 - val_loss: 2.8565 - val_accuracy: 0.6147
Epoch 73/100
64/64 [==============================] - 24s 369ms/step - loss: 0.0019 - accuracy:
0.9989 - val_loss: 3.6815 - val_accuracy: 0.6301
Epoch 74/100
64/64 [==============================] - 23s 364ms/step - loss: 0.0085 - accuracy:
0.9969 - val_loss: 2.9839 - val_accuracy: 0.6621
Epoch 75/100
64/64 [==============================] - 23s 360ms/step - loss: 0.0068 - accuracy:
0.9979 - val_loss: 3.3570 - val_accuracy: 0.4951
Epoch 76/100
64/64 [==============================] - 24s 368ms/step - loss: 3.8937e-04 - accuracy:
1.0000 - val_loss: 4.5387 - val_accuracy: 0.5044
Epoch 77/100
64/64 [==============================] - 23s 364ms/step - loss: 0.0011 - accuracy:
0.9999 - val_loss: 3.2457 - val_accuracy: 0.5774
Epoch 78/100
64/64 [==============================] - 24s 368ms/step - loss: 0.0044 - accuracy:
0.9984 - val_loss: 4.0395 - val_accuracy: 0.5518
Epoch 79/100
64/64 [==============================] - 24s 368ms/step - loss: 0.0123 - accuracy:
0.9961 - val_loss: 3.6527 - val_accuracy: 0.6245
Epoch 80/100
64/64 [==============================] - 24s 372ms/step - loss: 0.0215 - accuracy:
```

```
0.9952 - val_loss: 3.2264 - val_accuracy: 0.6084
Epoch 81/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0050 - accuracy:
0.9984 - val_loss: 2.8905 - val_accuracy: 0.6245
Epoch 82/100
64/64 [==============================] - 23s 355ms/step - loss: 0.0186 - accuracy:
0.9940 - val_loss: 1.5342 - val_accuracy: 0.6626
Epoch 83/100
64/64 [==============================] - 23s 365ms/step - loss: 0.0036 - accuracy:
0.9995 - val_loss: 3.9380 - val_accuracy: 0.6257
Epoch 84/100
64/64 [==============================] - 23s 363ms/step - loss: 0.0040 - accuracy:
0.9982 - val_loss: 5.4605 - val_accuracy: 0.5557
Epoch 85/100
64/64 [==============================] - 24s 368ms/step - loss: 0.0044 - accuracy:
0.9978 - val_loss: 5.9055 - val_accuracy: 0.6460
Epoch 86/100
64/64 [==============================] - 23s 366ms/step - loss: 0.0069 - accuracy:
0.9983 - val_loss: 3.4676 - val_accuracy: 0.6064
Epoch 87/100
64/64 [==============================] - 23s 362ms/step - loss: 0.0048 - accuracy:
0.9988 - val_loss: 4.5562 - val_accuracy: 0.6016
Epoch 88/100
64/64 [==============================] - 23s 362ms/step - loss: 2.4601e-04 - accuracy:
1.0000 - val_loss: 4.7289 - val_accuracy: 0.6284
Epoch 89/100
64/64 [==============================] - 23s 366ms/step - loss: 9.5441e-04 - accuracy:
0.9996 - val_loss: 5.0640 - val_accuracy: 0.6128
Epoch 90/100
64/64 [==============================] - 24s 372ms/step - loss: 1.0427e-04 - accuracy:
1.0000 - val_loss: 4.9349 - val_accuracy: 0.6064
Epoch 91/100
64/64 [==============================] - 23s 365ms/step - loss: 1.6246e-04 - accuracy:
1.0000 - val_loss: 5.4669 - val_accuracy: 0.5923
Epoch 92/100
64/64 [==============================] - 23s 365ms/step - loss: 1.7083e-04 - accuracy:
1.0000 - val_loss: 4.8902 - val_accuracy: 0.5562
Epoch 93/100
64/64 [==============================] - 24s 370ms/step - loss: 0.0039 - accuracy:
0.9987 - val_loss: 4.0293 - val_accuracy: 0.5649
Epoch 94/100
64/64 [==============================] - 23s 365ms/step - loss: 0.0043 - accuracy:
0.9985 - val_loss: 2.7171 - val_accuracy: 0.5532
Epoch 95/100
64/64 [==============================] - 23s 365ms/step - loss: 0.0024 - accuracy:
0.9987 - val_loss: 4.3506 - val_accuracy: 0.6382
Epoch 96/100
64/64 [==============================] - 24s 371ms/step - loss: 0.0112 - accuracy:
0.9962 - val_loss: 6.4583 - val_accuracy: 0.5149
Epoch 97/100
64/64 [==============================] - 24s 375ms/step - loss: 0.0140 - accuracy:
0.9957 - val_loss: 3.3490 - val_accuracy: 0.6118
Epoch 98/100
64/64 [==============================] - 24s 371ms/step - loss: 0.0018 - accuracy:
0.9994 - val_loss: 5.1457 - val_accuracy: 0.5537
Epoch 99/100
64/64 [==============================] - 24s 372ms/step - loss: 0.0029 - accuracy:
0.9990 - val_loss: 4.5643 - val_accuracy: 0.6035
Epoch 100/100
64/64 [==============================] - 24s 370ms/step - loss: 5.7734e-04 - accuracy:
1.0000 - val_loss: 4.7228 - val_accuracy: 0.5869
<tensorflow.python.keras.callbacks.History at 0x7f530c47d4d0>
```