

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

```
! wget https://www.dropbox.com/sh/s9r1av3m4eatd3y/AAA8zYti5b5tnyKfcah2Reaja -O data
```

```
--2023-07-09 14:29:15-- https://www.dropbox.com/sh/s9r1av3m4eatd3y/AAA8zYti5b5tnyKfcah2Reaja
Resolving www.dropbox.com (www.dropbox.com)... 162.125.6.18, 2620:100:6018:18::a27d:312
Connecting to www.dropbox.com (www.dropbox.com)|162.125.6.18|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: /sh/raw/s9r1av3m4eatd3y/AAA8zYti5b5tnyKfcah2Reaja [following]
--2023-07-09 14:29:15-- https://www.dropbox.com/sh/raw/s9r1av3m4eatd3y/AAA8zYti5b5tnyKfcah2Reaja
Reusing existing connection to www.dropbox.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://uc078d4f84696633152e9d0e7228.dl.dropboxusercontent.com/zip_download_get/Bi7k6RaGQ0Syt_v6As9XfBT
--2023-07-09 14:29:16-- https://uc078d4f84696633152e9d0e7228.dl.dropboxusercontent.com/zip_download_get/Bi7k6RaG
Resolving uc078d4f84696633152e9d0e7228.dl.dropboxusercontent.com (uc078d4f84696633152e9d0e7228.dl.dropboxusercontent.com)
Connecting to uc078d4f84696633152e9d0e7228.dl.dropboxusercontent.com (uc078d4f84696633152e9d0e7228.dl.dropboxusercontent.com)
HTTP request sent, awaiting response... 200 OK
Length: 108475527 (103M) [application/zip]
Saving to: 'data'

data          100%[=====>] 103.45M  24.2MB/s   in 4.3s

2023-07-09 14:29:21 (24.2 MB/s) - 'data' saved [108475527/108475527]
```

```
!unzip data -d "images/"
```

```
Archive: data
warning: stripped absolute path spec from /
mapname: conversion of failed
replace images/Train/Pikachu/345.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

```
pip install keras_preprocessing
```

```
Collecting keras_preprocessing
  Downloading Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
    42.6/42.6 kB 2.2 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.10/dist-packages (from keras_preprocessing)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from keras_preprocessing) (
Installing collected packages: keras_preprocessing
Successfully installed keras_preprocessing-1.1.2
```

```
classes = os.listdir("images/Train")
```

```
classes
```

```
['Charmander', 'Pikachu', 'Bulbasaur']
```

```
for pokemon_type in classes:
```

```
    path = "images/Train/" + pokemon_type
    print(f"{pokemon_type} - {len(os.listdir(path))}")
```

```
Charmander - 102
Pikachu - 101
Bulbasaur - 101
```

```
for pokemon_type in classes:
    path = "images/Test/" + pokemon_type
    print(f"{pokemon_type} - {len(os.listdir(path))}")
```

```
Charmander - 42
Pikachu - 40
Bulbasaur - 41
```

```
from keras_preprocessing import image
```

```
train_data = []
train_labels = []
```

```
for category in classes:
    folder = f"images/Train/{category}"
    for image_name in os.listdir(folder):
        img_path = os.path.join(folder, image_name)

        img = image.load_img(img_path, target_size = (100,100))
        img_array = image.img_to_array(img)
        train_data.append(img_array)
        train_labels.append(category)
```

```
/usr/local/lib/python3.10/dist-packages/PIL/Image.py:975: UserWarning: Palette images with Transparency expressed
warnings.warn(
```



```
len(train_data)
```

```
304
```

```
train_data = np.array(train_data)
train_labels = np.array(train_labels)
```

```
train_data = train_data.reshape(len(train_data), 30000)
```

```
train_data.shape
```

```
(304, 30000)
```

```
category2label = {"Pikachu": 0, "Charmander": 1, "Bulbasaur": 2}
label2category = {0: "Pikachu", 1: "Charmander", 2: "Bulbasaur"}
```

```
train_labels = np.array([category2label[label] for label in train_labels])
```

```
from keras.utils import to_categorical
```

```
train_labels = to_categorical(train_labels)
```

```
train_labels
```

```
array([[0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.]])
```

```
from keras.layers import Dense
from keras import Sequential
```

```
model = Sequential()
```

```
model.add(Dense(units = 256, activation = 'relu'))
```

```
model.add(Dense(units = 64, activation = 'relu'))
```

```
model.compile(optimizer = "adam", loss = "categorical_crossentropy", metrics = ["accuracy"])
```

```
model.summary()
```

```
Model: "sequential"
```

| Layer (type) | Output Shape | Param # |
|------------------------------|--------------|----------|
| dense (Dense) | (None, 512) | 15360512 |
| dense_1 (Dense) | (None, 512) | 262656 |
| dense_2 (Dense) | (None, 256) | 131328 |
| dense_3 (Dense) | (None, 128) | 32896 |
| dense_4 (Dense) | (None, 64) | 8256 |
| dense_6 (Dense) | (None, 3) | 195 |
| dense_7 (Dense) | (None, 512) | 2048 |
| dense_8 (Dense) | (None, 256) | 131328 |
| dense_9 (Dense) | (None, 128) | 32896 |
| dense_10 (Dense) | (None, 64) | 8256 |
| dense_11 (Dense) | (None, 3) | 195 |
| Total params: 15,970,566 | | |
| Trainable params: 15,970,566 | | |
| Non-trainable params: 0 | | |

```
model.fit(train_data, train_labels, batch_size = 32, epochs = 50)
```

```
Epoch 1/50
10/10 [=====] - 6s 344ms/step - loss: 1.0992 - accuracy: 0.3355
Epoch 2/50
10/10 [=====] - 3s 343ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 3/50
10/10 [=====] - 3s 303ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 4/50
10/10 [=====] - 4s 440ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 5/50
10/10 [=====] - 3s 298ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 6/50
10/10 [=====] - 3s 266ms/step - loss: 1.0988 - accuracy: 0.3355
Epoch 7/50
10/10 [=====] - 3s 299ms/step - loss: 1.0986 - accuracy: 0.3355
Epoch 8/50
10/10 [=====] - 4s 437ms/step - loss: 1.0986 - accuracy: 0.3355
Epoch 9/50
10/10 [=====] - 3s 310ms/step - loss: 1.0988 - accuracy: 0.3355
Epoch 10/50
10/10 [=====] - 3s 322ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 11/50
10/10 [=====] - 3s 266ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 12/50
10/10 [=====] - 4s 442ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 13/50
10/10 [=====] - 3s 344ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 14/50
10/10 [=====] - 3s 344ms/step - loss: 1.0986 - accuracy: 0.3355
Epoch 15/50
10/10 [=====] - 4s 414ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 16/50
10/10 [=====] - 4s 346ms/step - loss: 1.0986 - accuracy: 0.3355
Epoch 17/50
10/10 [=====] - 3s 282ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 18/50
10/10 [=====] - 3s 342ms/step - loss: 1.0987 - accuracy: 0.2961
Epoch 19/50
10/10 [=====] - 4s 408ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 20/50
10/10 [=====] - 3s 289ms/step - loss: 1.0987 - accuracy: 0.2829
```

```

Epoch 21/50
10/10 [=====] - 3s 269ms/step - loss: 1.0988 - accuracy: 0.3355
Epoch 22/50
10/10 [=====] - 3s 277ms/step - loss: 1.0986 - accuracy: 0.3355
Epoch 23/50
10/10 [=====] - 3s 350ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 24/50
10/10 [=====] - 4s 358ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 25/50
10/10 [=====] - 3s 274ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 26/50
10/10 [=====] - 3s 270ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 27/50
10/10 [=====] - 4s 391ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 28/50
10/10 [=====] - 3s 336ms/step - loss: 1.0987 - accuracy: 0.3355
Epoch 29/50
10/10 [=====] - 3s 315ms/step - loss: 1.0987 - accuracy: 0.3355

```

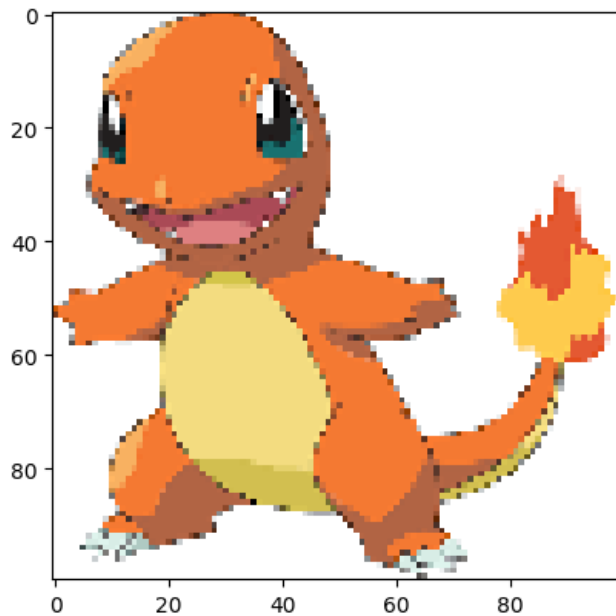
```

test_image = "/content/download.png"
img = image.load_img(test_image, target_size = (100, 100))
img = image.img_to_array(img)
img.shape

(100, 100, 3)

```

```
plt.imshow(img.astype('int'));
```



```

img = img.reshape(1, 30000)

model.predict(img).argmax()

1/1 [=====] - 0s 142ms/step
1

pred = label2category[model.predict(img).argmax()]
pred

1/1 [=====] - 0s 53ms/step
'Charmander'

```

