```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator

IMG_SIZE = 244
BATCH_SIZE = 32

train_datagen = ImageDataGenerator(rescale=1./255,
validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/pokemon',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='training'
)
val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/pokemon',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='validation'
)
```

```
Found 648 images belonging to 2 classes.
Found 161 images belonging to 2 classes.
```

```
# Define the model
model = keras.Sequential([
    layers.Conv2D(32,
(3,3),activation='relu',input_shape=(IMG_SIZE,IMG_SIZE,3)),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(128,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Flatten(),
    layers.Dense(128,activation='relu'),
    layers.Dense(1,activation='sigmoid') #output layer
])

# compile the model
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['ac
curacy'])

model.fit(train_generator, epochs=5, validation_data=val_generator)
```

```
Epoch 1/5
21/21 [==============================] - 208s 10s/step - loss: 0.0084
- accuracy: 0.9969 - val_loss: 1.8890e-38 - val_accuracy: 1.0000
```

```
Epoch 2/5
21/21 [==============================] - 104s 5s/step - loss:
0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy:
1.0000
Epoch 3/5
21/21 [==============================] - 108s 5s/step - loss:
0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy:
1.0000
Epoch 4/5
21/21 [==============================] - 116s 6s/step - loss:
0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy:
1.0000
Epoch 5/5
21/21 [==============================] - 103s 5s/step - loss:
0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy:
1.0000

<keras.src.callbacks.History at 0x7e2578522590>

model.save("pokemon.h5","label.txt")

/usr/local/lib/python3.10/dist-packages/keras/src/engine/
training.py:3103: UserWarning: You are saving your model as an HDF5
file via `model.save()`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')`.
  saving_api.save_model(

from tensorflow.keras.models import load_model
from tensorflow.kersa.preprocessing import image
import numpy as np


#load the save model
model = load_model('/content/pokemon.h5')

#load preprocessor the test image
test_image =
image.load_img('/content/drive/MyDrive/pokemon/images/abra.png',
target_size=(224, 224))
img = image.load_img(test_image, target_size=(224, 224))
img = image.img_to_array(test_image)
img = np.expand_dims(img, axis=0)
```