```
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/car datset/dataset/Images/Train',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)

val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/car datset/dataset/Images/Train',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

```
Found 52 images belonging to 3 classes.
Found 12 images belonging to 3 classes.
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

```
# 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(6,activation='sigmoid') #output layer
])

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

```
curacy'])


model.save("car dataset.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.keras.preprocessing import image
import numpy as np

model = load_model('/content/drive/MyDrive/car datset/car dataset.h5')
test_image_path ='/content/drive/MyDrive/car
datset/dataset/Images/Test/audi/21.jpg'
# Load and resize the image to match the model's input shape
img = image.load_img(test_image_path, target_size=(244, 244)) # Change
target size to 244x244
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)
img_array /= 255.
prediction = model.predict(img_array)
print(prediction)
```

```
1/1 [==============================] - 0s 124ms/step
[[1. 1. 1. 1. 1. 1.]]
```

```
# Assuming you want to check if ANY of the predictions are greater
than 0.5
if (prediction > 0.5).any():
    print('It is a  audicar')
else:
    print('It is a merced')
```

```
It is a  audicar
```