```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
IMG_SIZE = 224
BATCH_SIZE = 32

train_datagen = ImageDataGenerator(rescale=1./255,
validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/ML_TEAM5/1SV21CS046/Cars Dataset/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/ML_TEAM5/1SV
21CS046/Cars Dataset/train',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)

Found 1370 images belonging to 7 classes.
Found 341 images belonging to 7 classes.

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
])

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

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

Epoch 1/5
43/43 [==============================] - 199s 5s/step - loss: 0.4103 -
accuracy: 0.8571 - val_loss: 0.4102 - val_accuracy: 0.8571
Epoch 2/5
43/43 [==============================] - 200s 5s/step - loss: 0.4102 -
```

```
accuracy: 0.8571 - val_loss: 0.4102 - val_accuracy: 0.8571
Epoch 3/5
43/43 [==============================] - 204s 5s/step - loss: 0.4102 -
accuracy: 0.8571 - val_loss: 0.4102 - val_accuracy: 0.8571
Epoch 4/5
43/43 [==============================] - 192s 4s/step - loss: 0.4102 -
accuracy: 0.8571 - val_loss: 0.4102 - val_accuracy: 0.8571
Epoch 5/5
43/43 [==============================] - 182s 4s/step - loss: 0.4102 -
accuracy: 0.8571 - val_loss: 0.4102 - val_accuracy: 0.8571

<keras.src.callbacks.History at 0x7864f4261690>

model.save("Model.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/Model.h5")

test_image_path = '/content/drive/MyDrive/ML_TEAM5/1SV21CS046/Cars
Dataset/test/Toyota Innova/1002.jpg'
img = image.load_img(test_image_path,target_size=(224,224))
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 116ms/step
[[0.14484452]]

if prediction < 0.33:
  print("Toyota",prediction[0][0])
elif prediction < 0.66:
  print("Rolls Royce",prediction[0][0])

else:
  print("Mahindra", prediction[0][0])

Toyota 0.14484452
```