DARSHAN K
1SV21cs025

```
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
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/Cars
Dataset/train',     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/Cars
Dataset/train',     target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
class_mode='binary',
subset='validation'
)

Found 1423 images belonging to 5 classes.
Found 354 images belonging to 5 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')
])

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

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

```
Epoch 1/5
45/45 [==============================] - 170s 4s/step - loss: -
377756024832.0000 - accuracy: 0.1525 - val_loss: -612959256576.0000 -
val_accuracy: 0.1525
Epoch 2/5
45/45 [==============================] - 171s 4s/step - loss:
999186825216.0000 - accuracy: 0.1525 - val_loss: -1504027607040.0000 -
val_accuracy: 0.1525
Epoch 3/5
45/45 [==============================] - 178s 4s/step - loss: -
2230093348864.0000 - accuracy: 0.1525 - val_loss: -3242398056448.0000
- val_accuracy: 0.1525
Epoch 4/5
45/45 [==============================] - 176s 4s/step - loss: -
4542287052800.0000 - accuracy: 0.1525 - val_loss: -6259808927744.0000
- val_accuracy: 0.1525
Epoch 5/5
45/45 [==============================] - 168s 4s/step - loss: -
8445961961472.0000 - accuracy: 0.1525 - val_loss: -11295716802560.0000
- val_accuracy: 0.1525 <keras.src.callbacks.History at 0x78a6387a8bb0>
```

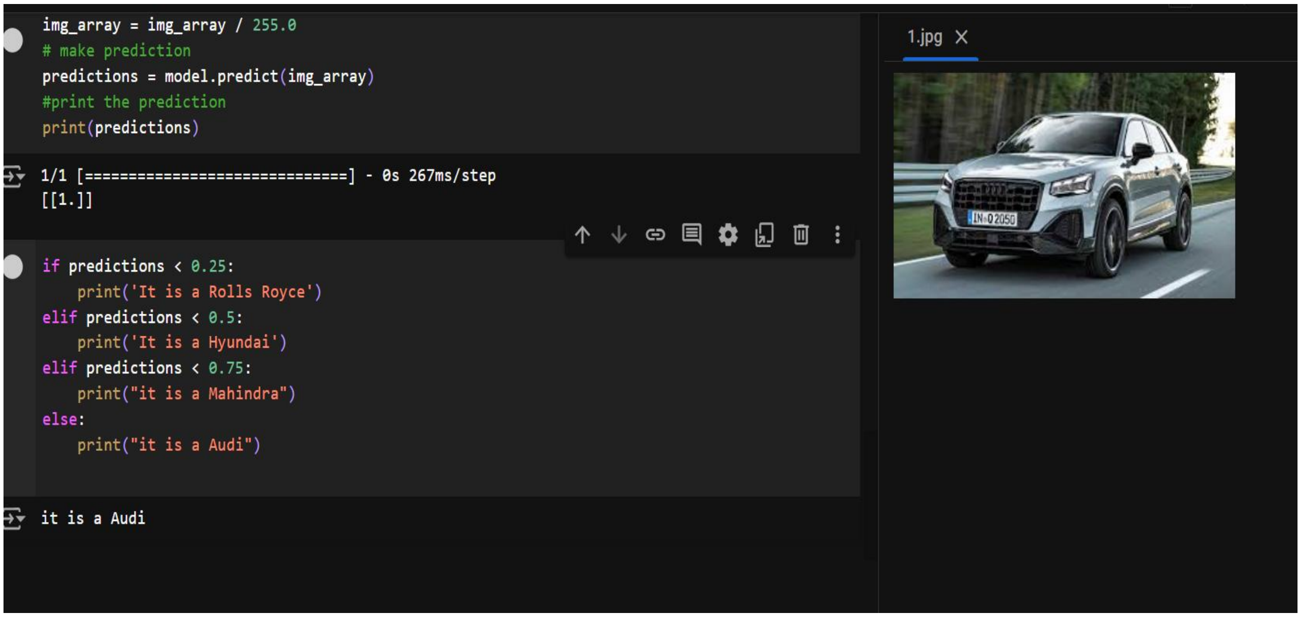
```
  model.save("model.h5","label.txt")
```

```
/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.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np # Load the saved model
model = load_model('/content/model.h5') # Load
and preprocessor the test image
test_image_path = '/content/drive/MyDrive/Cars
Dataset/train/Audi/1.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)
# add batch demension
img_array = img_array / 255.0
# make prediction
predictions = model.predict(img_array)
#print the prediction print(predictions)
```

```
1/1 [==============================] - 0s 267ms/step
[[1.]]

if predictions < 0.25:
    print('It is a Rolls Royce')
elif predictions < 0.5:
    print('It is a Hyundai')
elif predictions < 0.75:
    print("it is a Mahindra")
else:     print("it is a
Audi") it is a Audi
```



```
img_array = img_array / 255.0
# make prediction
predictions = model.predict(img_array)
#print the prediction
print(predictions)
```

```
1/1 [==============================] - 0s 267ms/step
[[1.]]
```

```
if predictions < 0.25:
    print('It is a Rolls Royce')
elif predictions < 0.5:
    print('It is a Hyundai')
elif predictions < 0.75:
    print("it is a Mahindra")
else:
    print("it is a Audi")
```

```
it is a Audi
```