

day1-003-1

June 26, 2024

```
[ ]: 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 = 244
BATCH_SIZE = 32
```

```
[ ]: train_datagen = ImageDataGenerator(rescale=1./255,validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/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/Train',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

Found 502 images belonging to 2 classes.

Found 125 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=['accuracy'])

```

```

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

```

```

Epoch 1/5
16/16 [=====] - 182s 11s/step - loss: 1.1621 -
accuracy: 0.5000 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 2/5
16/16 [=====] - 74s 5s/step - loss: 0.6932 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 3/5
16/16 [=====] - 76s 5s/step - loss: 0.6931 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 4/5
16/16 [=====] - 74s 5s/step - loss: 0.6931 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 5/5
16/16 [=====] - 77s 5s/step - loss: 0.6931 - accuracy:
0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000

```

```

[ ]: <keras.src.callbacks.History at 0x7c936722ca60>

```

```

[ ]: 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/Train/model.h5")

test_image_path = '/content/drive/MyDrive/Train/Rolls Royce/26.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 (244, 244)
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 120ms/step
[[0.5005738]]

```

```

[ ]: if prediction < 0.5:
    print('It is a Mahindra Scorpio')
else:
    print('It is a Rolls Royce')

```

It is a Rolls Royce