

Assignment 4

In [1]:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [2]:

```
train_datagen = ImageDataGenerator(rescale=1/255, zoom_range=0.2, horizontal_flip=True, vertic
```

In [3]:

```
train_datagen = ImageDataGenerator(rescale=1/255)
```

In [4]:

```
test_datagen = ImageDataGenerator(rescale=1/255)
```

In [5]:

```
n_directory(r'E:\AIML_Externship\Contents\Cars_Dataset\train', target_size=(64,64), class_mode
```

Found 3352 images belonging to 7 classes.

In [6]:

```
len(x_train)
```

Out[6]:

34

In [7]:

```
m_directory(r'E:\AIML_Externship\Contents\Cars_Dataset\test', target_size=(64,64), class_mode
```

Found 813 images belonging to 7 classes.

In [8]:

```
len(x_test)
```

Out[8]:

9

In [9]:

```
x_train.class_indices
```

Out[9]:

```
{'Audi': 0,  
 'Hyundai Creta': 1,  
 'Mahindra Scorpio': 2,  
 'Rolls Royce': 3,  
 'Swift': 4,  
 'Tata Safari': 5,  
 'Toyota Innova': 6}
```

Import libraries

In [10]:

```
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

Create the model

In [11]:

```
model = Sequential()
```

Add layers

In [12]:

```
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

In [13]:

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [14]:

```
model.add(Flatten())
```

In [15]:

```
#hidden layer - 1  
model.add(Dense(300,activation = 'relu'))
```

In [16]:

```
#hiddenlayer - 2  
model.add(Dense(150,activation='relu'))
```

In [17]:

```
#output layer  
model.add(Dense(7,activation='softmax'))
```

Compile the model

In [18]:

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

Fit the model

In [19]:

```
#model.fit_generator(x_train,steps_per_epoch = len(x_train),epochs=10,validation_data=x_test)

#model.fit_generator(x_train,steps_per_epoch = len(x_train),epochs=10,validation_data = x_test)

#model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10)

model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,v
```

Epoch 1/10

C:\Users\RAJVEE~1\AppData\Local\Temp\ipykernel_15588\1177999562.py:7: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

```
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

```
34/34 [=====] - 10s 281ms/step - loss: 2.0916 - accuracy: 0.2730 - val_loss: 1.7049 - val_accuracy: 0.3801
```

Epoch 2/10

```
34/34 [=====] - 9s 270ms/step - loss: 1.4838 - accuracy: 0.4896 - val_loss: 1.4725 - val_accuracy: 0.4748
```

Epoch 3/10

```
34/34 [=====] - 9s 262ms/step - loss: 1.0609 - accuracy: 0.6450 - val_loss: 1.2528 - val_accuracy: 0.5658
```

Epoch 4/10

```
34/34 [=====] - 9s 275ms/step - loss: 0.7164 - accuracy: 0.7780 - val_loss: 1.1533 - val_accuracy: 0.6064
```

Epoch 5/10

```
34/34 [=====] - 10s 286ms/step - loss: 0.4441 - accuracy: 0.8857 - val_loss: 1.1966 - val_accuracy: 0.6162
```

Epoch 6/10

```
34/34 [=====] - 8s 243ms/step - loss: 0.2545 - accuracy: 0.9445 - val_loss: 1.2471 - val_accuracy: 0.6187
```

Epoch 7/10

```
34/34 [=====] - 9s 267ms/step - loss: 0.1386 - accuracy: 0.9726 - val_loss: 1.4756 - val_accuracy: 0.6199
```

Epoch 8/10

```
34/34 [=====] - 9s 265ms/step - loss: 0.0715 - accuracy: 0.9937 - val_loss: 1.4703 - val_accuracy: 0.6261
```

Epoch 9/10

```
34/34 [=====] - 9s 276ms/step - loss: 0.0283 - accuracy: 0.9991 - val_loss: 1.5723 - val_accuracy: 0.6212
```

Epoch 10/10

```
34/34 [=====] - 9s 279ms/step - loss: 0.0148 - accuracy: 1.0000 - val_loss: 1.5954 - val_accuracy: 0.6298
```

Out[19]:

```
<keras.callbacks.History at 0x1fd08140fa0>
```

In [41]:

```
model.save('cars.h5')
```

Testing the CNN model

In [42]:

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

In [43]:

```
model = load_model('cars.h5')
```

In [23]:

```
img = image.load_img(r'E:\AIML_Externship\Contents\Cars_Dataset\test\Rolls Royce\101.jpg',t
```

In [24]:

```
img
```

Out[24]:



In [25]:

```
x=image.img_to_array(img)
```

In [26]:

```
x.ndim
```

Out[26]:

```
3
```

In [27]:

```
x=np.expand_dims(x,axis=0)
```

In [28]:

x

Out[28]:

```

array([[[[ 6.,  5., 13.],
          [16., 15., 23.],
          [18., 17., 25.],
          ...,
          [160., 172., 188.],
          [159., 171., 187.],
          [229., 232., 237.]],

        [[ 6.,  5., 13.],
          [22., 21., 29.],
          [36., 35., 43.],
          ...,
          [152., 165., 181.],
          [154., 167., 183.],
          [190., 188., 191.]],

        [[ 46., 45., 53.],
          [52., 51., 59.],
          [51., 50., 58.],
          ...,
          [159., 175., 190.],
          [164., 180., 195.],
          [160., 172., 186.]],

        ...,

        [[ 78., 76., 79.],
          [76., 74., 77.],
          [77., 75., 78.],
          ...,
          [100., 98., 99.],
          [98., 96., 97.],
          [94., 92., 93.]],

        [[ 74., 74., 76.],
          [73., 73., 75.],
          [75., 75., 77.],
          ...,
          [96., 94., 95.],
          [99., 97., 98.],
          [101., 95., 97.]],

        [[ 70., 70., 72.],
          [69., 69., 71.],
          [75., 75., 77.],
          ...,
          [99., 97., 98.],
          [93., 91., 92.],
          [97., 93., 94.] ]]], dtype=float32)

```

In [29]:

```
x.ndim
```

Out[29]:

4

In [30]:

```
pred=np.argmax(model.predict(x),axis=1)
```

In [31]:

```
pred
```

Out[31]:

```
array([6], dtype=int64)
```

In [32]:

```
index = ['Audi','Hyundai Creta','Mahindra Scorpio','Rolls Royce','Swift','Tata Safari','Toy  
print(index[pred[0]])
```

Toyota Innova

Open CV

In [33]:

```
import cv2
```

In [34]:

```
img = cv2.imread(r'E:\AIML_Externship\Contents\Cars_Dataset\test\Rolls Royce\101.jpg',1)
```

In [35]:

img

Out[35]:

```

array([[ 7,  0,  0],
       [ 7,  0,  0],
       [ 9,  1,  2],
       ...,
       [242, 234, 227],
       [235, 231, 226],
       [231, 227, 222]],

       [[ 10,  2,  3],
       [ 11,  3,  4],
       [ 13,  5,  6],
       ...,
       [239, 233, 228],
       [234, 231, 227],
       [231, 226, 223]],

       [[ 14,  6,  7],
       [ 14,  6,  7],
       [ 15,  7,  8],
       ...,
       [228, 221, 218],
       [221, 219, 218],
       [221, 217, 216]],

       ...,

       [[ 72,  70,  70],
       [ 72,  70,  70],
       [ 72,  70,  70],
       ...,
       [ 97,  96, 100],
       [ 96,  95,  99],
       [ 96,  95,  99]],

       [[ 73,  71,  71],
       [ 72,  70,  70],
       [ 72,  70,  70],
       ...,
       [ 94,  93,  97],
       [ 94,  93,  97],
       [ 94,  93,  97]],

       [[ 74,  72,  72],
       [ 74,  72,  72],
       [ 74,  72,  72],
       ...,
       [ 91,  90,  94],
       [ 91,  90,  94],
       [ 90,  89,  93]]], dtype=uint8)

```

In [36]:

```
img1 = cv2.imread(r'E:\AIML_Externship\Contents\Cars_Dataset\test\Rolls Royce\101.jpg',0)
```


In [37]:

img1

Out[37]:

```
array([[ 0,  0,  2, ..., 233, 230, 226],
       [ 3,  4,  6, ..., 232, 230, 226],
       [ 7,  7,  8, ..., 221, 219, 217],
       ...,
       [70, 70, 70, ..., 97, 96, 96],
       [71, 70, 70, ..., 94, 94, 94],
       [72, 72, 72, ..., 91, 91, 90]], dtype=uint8)
```

In [38]:

```
print(img.shape)
```

(183, 275, 3)

In [39]:

```
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

CNN Video Analysis

In [40]:

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('cars.h5')
video=cv2.VideoCapture(0)
index=['Audi', 'Hyundai Creta', 'Mahindra Scorpio', 'Rolls Royce', 'Swift', 'Tata Safari', 'Toyota']
while 1:
    succes,frame=video.read()
    cv2.imwrite('image.jpg',frame)
    img=image.load_img('image.jpg',target_size=(64,64))
    x=image.img_to_array(img)
    x=np.expand_dims(x,axis=0)
    pred=np.argmax(model.predict(x),axis=1)
    y=pred[0]
    cv2.putText(frame,'The predicted Cars is: '+str(index[y]),(100,100),cv2.FONT_HERSHEY_SIMPLEX,1)
    cv2.imshow('image',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
video.release()
cv2.destroyAllWindows()
```

In []:

