**CODE FOR IMAGE CLASSIFICATION:**

# 1. Import libraries from Keras.  
import tensorflow as tf  
  
from keras.layers import Conv2D  
from keras.layers import Dense  
from keras.layers import Flatten  
from keras.layers import MaxPooling2D  
from keras.models import Sequential  
  
# 2. Configure the CNN (Convolutional Neural Network).  
classifier = Sequential()  
# 2a. Convolution - extracting appropriate features from the input image.  
# Non-Linearity (RELU) - replacing all negative pixel values in feature map by zero.  
classifier.add(Conv2D(32, 3, 3, input\_shape=(64, 64, 3), activation='relu'))  
# 2b. Pooling: reduces dimensionality of the feature maps but keeps the most important information.  
classifier.add(MaxPooling2D(pool\_size=(2, 2)))  
# 2c. Adding a second convolutional layer and flattening in order to arrange 3D volumes into a 1D vector.  
#classifier.add(Conv2D(32, (3, 3), activation='relu'))  
#classifier.add(MaxPooling2D(pool\_size=(2, 2)))  
classifier.add(Flatten())  
  
# 2d. Fully connected layers: ensures connections to all activations in the previous layer.  
classifier.add(Dense(output\_dim=128, activation='relu'))  
classifier.add(Dense(output\_dim=1, activation='sigmoid'))  
  
# 3. Compile the CNN and train the classifier..  
classifier.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])  
  
from keras.preprocessing.image import ImageDataGenerator  
  
train\_datagen = ImageDataGenerator(  
 rescale=1. / 255,  
 shear\_range=0.2,  
 zoom\_range=0.2,  
 horizontal\_flip=True)  
  
test\_datagen = ImageDataGenerator(rescale=1. / 255)  
  
training\_set = train\_datagen.flow\_from\_directory(  
 'C:/Users/Kiran Lalwani/Desktop/dss/training\_set',  
 target\_size=(64, 64),  
 batch\_size=32,  
 class\_mode='binary')  
  
test\_set = test\_datagen.flow\_from\_directory(  
 'C:/Users/Kiran Lalwani/Desktop/dss/test\_set',  
 target\_size=(64, 64),  
 batch\_size=32,  
 class\_mode='binary')  
  
from IPython.display import display  
from PIL import Image  
  
classifier.fit\_generator(  
 training\_set,  
 steps\_per\_epoch=22,  
 epochs=1,  
 validation\_data=test\_set,  
 validation\_steps=80)  
# 5. Generate predictions  
import numpy as np  
from keras.preprocessing import image  
test\_image = image.load\_img('C:/Users/Kiran Lalwani/Desktop/dss/sample-passport.jpg', target\_size=(64, 64))  
test\_image = image.img\_to\_array(test\_image)  
test\_image = np.expand\_dims(test\_image, axis=0)  
result = classifier.predict(test\_image)  
training\_set.class\_indices  
if result[0][0] >= 0.5:  
 prediction = 'passport'  
else:  
 prediction = 'license'  
print(prediction)

**SAMPLE PASSPORT:**

****

**OUTPUT:**

"C:\Users\Kiran Lalwani\PycharmProjects\cnn\venv\Scripts\python.exe" "C:/Users/Kiran Lalwani/PycharmProjects/cnn/test3.py"

Using TensorFlow backend.

C:/Users/Kiran Lalwani/PycharmProjects/cnn/test3.py:14: UserWarning: Update your `Conv2D` call to the Keras 2 API: `Conv2D(32, (3, 3), input\_shape=(64, 64, 3..., activation="relu")`

classifier.add(Conv2D(32, 3, 3, input\_shape=(64, 64, 3), activation='relu'))

C:/Users/Kiran Lalwani/PycharmProjects/cnn/test3.py:23: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(activation="relu", units=128)`

classifier.add(Dense(output\_dim=128, activation='relu'))

C:/Users/Kiran Lalwani/PycharmProjects/cnn/test3.py:24: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(activation="sigmoid", units=1)`

classifier.add(Dense(output\_dim=1, activation='sigmoid'))

Found 44 images belonging to 2 classes.

Found 14 images belonging to 2 classes.

Epoch 1/1

2019-01-06 10:21:53.079807: I tensorflow/core/platform/cpu\_feature\_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2

1/22 [>.............................] - ETA: 2:52 - loss: 0.7075 - acc: 0.5312

2/22 [=>............................] - ETA: 1:24 - loss: 2.1323 - acc: 0.5573

3/22 [===>..........................] - ETA: 55s - loss: 1.7145 - acc: 0.5278

4/22 [====>.........................] - ETA: 40s - loss: 1.6887 - acc: 0.5000

5/22 [=====>........................] - ETA: 36s - loss: 1.6779 - acc: 0.4812

6/22 [=======>......................] - ETA: 30s - loss: 1.5285 - acc: 0.5260

7/22 [========>.....................] - ETA: 27s - loss: 1.4612 - acc: 0.5089

8/22 [=========>....................] - ETA: 24s - loss: 1.3528 - acc: 0.5391

9/22 [===========>..................] - ETA: 22s - loss: 1.2797 - acc: 0.5312

10/22 [============>.................] - ETA: 19s - loss: 1.2200 - acc: 0.5198

11/22 [==============>...............] - ETA: 18s - loss: 1.1714 - acc: 0.5152

12/22 [===============>..............] - ETA: 16s - loss: 1.1276 - acc: 0.5208

13/22 [================>.............] - ETA: 14s - loss: 1.0919 - acc: 0.5192

14/22 [==================>...........] - ETA: 12s - loss: 1.0608 - acc: 0.5179

15/22 [===================>..........] - ETA: 11s - loss: 1.0328 - acc: 0.5333

16/22 [====================>.........] - ETA: 9s - loss: 1.0079 - acc: 0.5469

17/22 [======================>.......] - ETA: 7s - loss: 0.9838 - acc: 0.5699

18/22 [=======================>......] - ETA: 6s - loss: 0.9644 - acc: 0.5799

19/22 [========================>.....] - ETA: 4s - loss: 0.9433 - acc: 0.5954

20/22 [==========================>...] - ETA: 2s - loss: 0.9275 - acc: 0.5990

21/22 [===========================>..] - ETA: 1s - loss: 0.9107 - acc: 0.6091

22/22 [==============================] - 51s 2s/step - loss: 0.8933 - acc: 0.6190 - val\_loss: 0.5953 - val\_acc: 0.7143

passport

Process finished with exit code 0