MAHENDRAENGINEERINGCOLLEGEFORWOMEN

NAME: SAROJINI DEVI B

SUB:IBM

REGNO:61419106054

ASIGNMENT-3

```
import numpy as

npimportpandasaspd


fromPILimportImageFilefrom

tqdm import tqdmimporth5py

importcv2


importmatplotlib.pyplotasplt

%matplotlibinline


importseabornassns


fromsklearn.model_selectionimporttrain_test_splitfroms

klearn.metricsimportconfusion_matrix

fromsklearn.metricsimportplot_confusion_matrix


fromtensorflow.keras.utilsimportto_categorical
```

```
fromtensorflow.keras.preprocessingimportimageaskeras_imagefrom
tensorflow.keras.modelsimportSequential,load_model
fromtensorflow.keras.layersimportDense
fromtensorflow.keras.layersimportActivation,Dropout
fromtensorflow.keras.layersimportConv2D,MaxPooling2D,GlobalMaxPoling2Dfrom
tensorflow.keras.callbacks import ReduceLROnPlateau,
ModelCheckpointfromtensorflow.keras.layersimportLeakyReLU
defmodel():
model=Sequential()

model.add(Conv2D(128,(3,3),input_shape=x_train.shape[1:]))mode
l.add(LeakyReLU(alpha=0.02))

model.add(MaxPoling2D(pol_size=(2,2)))model.add(Dropout(0.2
5))

model.add(Conv2D(128, (3,
3)))model.add(LeakyReLU(alpha=0.02))

model.add(MaxPoling2D(pol_size=(2,2)))model.add(Dropout(0.2
5))

model.add(GlobalMaxPoling2D())

model.add(Dense(512))
```

```
model.add(LeakyReLU(alpha=0.02))model.add(Dro
pout(0.5))

model.add(Dense(10))model.
add(Activation('softmax'))

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

model=model()

#Tosavethebestmodel
checkpointer=ModelCheckpoint(filepath='weights.best.model.hdf5',verbose=2,save_best_on
ly=True)

#Toreducelearningratedynamically
lr_reduction = ReduceLROnPlateau(monitor='val_los ', patience=5, verbose=2,
factor=0.2)#Trainthemodel
history=model.fit(x_train,y_train,epochs=75,batch_size=32,verbose=2,validation_data=(x_vali
        d,y_valid),
        callbacks=[checkpointer,
data_generator=keras_image.ImageDataGenerator(shear_range=0.3,
                zoom_range=0.3,rotation_
                range=30,horizontal_flip=Tr
                ue)
```

```
dg_history=model.fit_generator(data_generator.flow(x_train,y_train,batch_size=64),steps_per
                               _epoch = len(x_train)// 64, epochs=7, verbose=2,validation_data=(x_valid,
                               y_valid),callbacks=[checkpointer,lr_reduction])
```