# EXP – 04 Face Recognition using Machine Learning CNN

**AIM:**

To develop a python on Face recognition using Machine learning CNN

**ALGORTIHM:**

Step 1: Getting Image dataset for our program

Step 2: Creating a mapping for index and face names

Step 3: Creating the CNN face recognition model

Step 4: Testing the CNN classifier on unseen images

**Program:**

TrainingImagePath='/Users/Joel/Python Case Studies/Face Images/Final Training Images'

from keras.preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator(

        shear\_range=0.1,

        zoom\_range=0.1,

        horizontal\_flip=True)

test\_datagen = ImageDataGenerator()

# Generating the Training Data

training\_set = train\_datagen.flow\_from\_directory(

        TrainingImagePath,

        target\_size=(64, 64),

        batch\_size=32,

        class\_mode='categorical')

# Generating the Testing Data

test\_set = test\_datagen.flow\_from\_directory(

        TrainingImagePath,

        target\_size=(64, 64),

        batch\_size=32,

        class\_mode='categorical')

# Printing class labels for each face

test\_set.class\_indices

# class\_indices have the numeric tag for each face

TrainClasses=training\_set.class\_indices

# Storing the face and the numeric tag for future reference

ResultMap={}

for faceValue,faceName in zip(TrainClasses.values(),TrainClasses.keys()):

    ResultMap[faceValue]=faceName

# Saving the face map for future reference

import pickle

with open("ResultsMap.pkl", 'wb') as fileWriteStream:

    pickle.dump(ResultMap, fileWriteStream)

# The model will give answer as a numeric tag

# This mapping will help to get the corresponding face name for it

print("Mapping of Face and its ID",ResultMap)

# The number of neurons for the output layer is equal to the number of faces

OutputNeurons=len(ResultMap)

print('\n The Number of output neurons: ', OutputNeurons)

from keras.models import Sequential

from keras.layers import Convolution2D

from keras.layers import MaxPool2D

from keras.layers import Flatten

from keras.layers import Dense

'''Initializing the Convolutional Neural Network'''

classifier= Sequential()

''' STEP--1 Convolution

# Adding the first layer of CNN

# we are using the format (64,64,3) because we are using TensorFlow backend

# It means 3 matrix of size (64X64) pixels representing Red, Green and Blue components of pixels

'''

classifier.add(Convolution2D(32, kernel\_size=(5, 5), strides=(1, 1), input\_shape=(64,64,3), activation='relu'))

'''# STEP--2 MAX Pooling'''

classifier.add(MaxPool2D(pool\_size=(2,2)))

classifier.add(Convolution2D(64, kernel\_size=(5, 5), strides=(1, 1), activation='relu'))

classifier.add(MaxPool2D(pool\_size=(2,2)))

classifier.add(Flatten())

classifier.add(Dense(64, activation='relu'))

classifier.add(Dense(OutputNeurons, activation='softmax'))

#classifier.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

classifier.compile(loss='categorical\_crossentropy', optimizer = 'adam', metrics=["accuracy"])

import time

# Measuring the time taken by the model to train

StartTime=time.time()

# Starting the model training

classifier.fit\_generator(

                    training\_set,

                    steps\_per\_epoch=30,

                    epochs=10,

                    validation\_data=test\_set,

                    validation\_steps=10)

EndTime=time.time()

print("###### Total Time Taken: ", round((EndTime-StartTime)/60), 'Minutes ######')

import numpy as np

from keras.preprocessing import image

ImagePath='/Users/Joel/Python Case Studies/Face Images/Final Testing Images/face4/3face4.jpg'

test\_image=image.load\_img(ImagePath,target\_size=(64, 64))

test\_image=image.img\_to\_array(test\_image)

test\_image=np.expand\_dims(test\_image,axis=0)

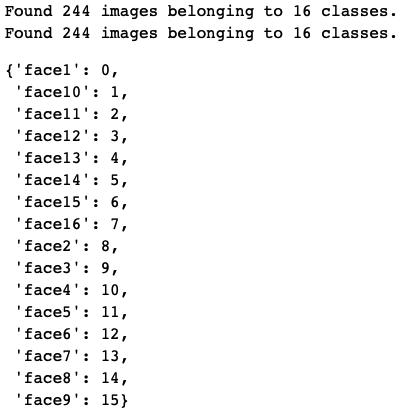
result=classifier.predict(test\_image,verbose=0)

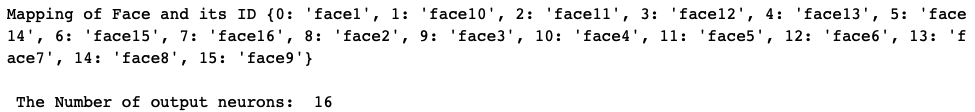
#print(training\_set.class\_indices)

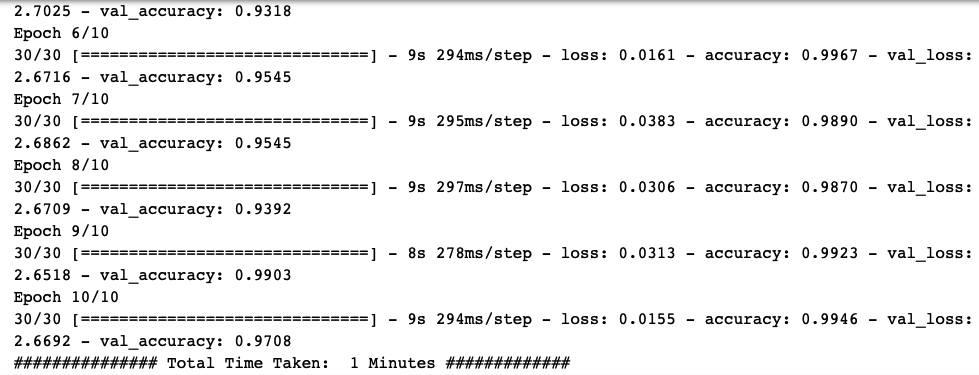
print('####'\*10)

print('Prediction is: ',ResultMap[np.argmax(result)])

**Output:**

****

****

****

****

**Result:**

Thus, a python on Face recognition using Machine learning CNN has been built successfully.