"""

Created on Sun Sep 22 14:43:34 2019

@author: Dheeraj A M

"""

import tensorflow as tf

import keras

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, AveragePooling2D

from keras.layers import Dense, Activation, Dropout, Flatten

from keras.preprocessing import image

from keras.preprocessing.image import ImageDataGenerator

import numpy as np

import matplotlib.pyplot as plt

num\_classes = 7

batch\_size = 256

epochs = 25

with open("fer2013.csv") as f:

content = f.readlines()

lines = np.array(content)

num\_of\_instances = lines.size

print("number of instances: ",num\_of\_instances)

print("instance length: ",len(lines[1].split(",")[1].split(" ")))

x\_train, y\_train, x\_test, y\_test = [], [], [], []

for i in range(1,num\_of\_instances):

try:

emotion, img, usage = lines[i].split(",")

val = img.split(" ")

pixels = np.array(val, 'float32')

emotion = keras.utils.to\_categorical(emotion, num\_classes)

if 'Training' in usage:

y\_train.append(emotion)

x\_train.append(pixels)

elif 'PublicTest' in usage:

y\_test.append(emotion)

x\_test.append(pixels)

except:

print("", end="")

x\_train = np.array(x\_train, 'float32')

y\_train = np.array(y\_train, 'float32')

x\_test = np.array(x\_test, 'float32')

y\_test = np.array(y\_test, 'float32')

x\_train /= 255

x\_test /= 255

x\_train = x\_train.reshape(x\_train.shape[0], 48, 48, 1)

x\_train = x\_train.astype('float32')

x\_test = x\_test.reshape(x\_test.shape[0], 48, 48, 1)

x\_test = x\_test.astype('float32')

print(x\_train.shape[0], 'train samples')

print(x\_test.shape[0], 'test samples')

model = Sequential()

model.add(Conv2D(64, (5, 5), activation='relu', input\_shape=(48,48,1)))

model.add(MaxPooling2D(pool\_size=(5,5), strides=(2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(AveragePooling2D(pool\_size=(3,3), strides=(2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(AveragePooling2D(pool\_size=(3,3), strides=(2, 2)))

model.add(Flatten())

model.add(Dense(1024, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(1024, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(num\_classes, activation='softmax'))

gen = ImageDataGenerator()

train\_generator = gen.flow(x\_train, y\_train, batch\_size=batch\_size)

model.compile(loss='categorical\_crossentropy'

, optimizer=keras.optimizers.Adam()

, metrics=['accuracy']

)

fit = False

if fit == True:

model.fit\_generator(train\_generator, steps\_per\_epoch=batch\_size, epochs=epochs) #train for randomly selected one

else:

model.load\_weights('model25.h5')