Apprentissage

import sys, os

import pandas as pd

import numpy as np

from keras.models import Sequential

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

from keras.layers import Conv2D, MaxPooling2D, BatchNormalization,AveragePooling2D

from keras.losses import categorical\_crossentropy

from keras.optimizers import Adam

from keras.regularizers import l2

from keras.utils import np\_utils

# pd.set\_option('display.max\_rows', 500)

# pd.set\_option('display.max\_columns', 500)

# pd.set\_option('display.width', 1000)

df=pd.read\_csv('fer2013.csv')

# print(df.info())

# print(df["Usage"].value\_counts())

# print(df.head())

X\_train,train\_y,X\_test,test\_y=[],[],[],[]

for index, row in df.iterrows():

    val=row['pixels'].split(" ")

    try:

        if 'Training' in row['Usage']:

           X\_train.append(np.array(val,'float32'))

           train\_y.append(row['emotion'])

        elif 'PublicTest' in row['Usage']:

           X\_test.append(np.array(val,'float32'))

           test\_y.append(row['emotion'])

    except:

        print(f"error occured at index :{index} and row:{row}")

num\_features = 64

num\_labels = 7

batch\_size = 64

epochs = 30

width, height = 48, 48

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

train\_y = np.array(train\_y,'float32')

X\_test = np.array(X\_test,'float32')

test\_y = np.array(test\_y,'float32')

train\_y=np\_utils.to\_categorical(train\_y, num\_classes=num\_labels)

test\_y=np\_utils.to\_categorical(test\_y, num\_classes=num\_labels)

#cannot produce

#normalizing data between oand 1

X\_train -= np.mean(X\_train, axis=0)

X\_train /= np.std(X\_train, axis=0)

X\_test -= np.mean(X\_test, axis=0)

X\_test /= np.std(X\_test, axis=0)

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

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

# print(f"shape:{X\_train.shape}")

##designing the cnn

#1st convolution layer

model = Sequential()

model.add(Conv2D(64, kernel\_size=(3, 3), activation='relu', input\_shape=(X\_train.shape[1:])))

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

# model.add(BatchNormalization())

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

model.add(Dropout(0.5))

#2nd convolution layer

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

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

# model.add(BatchNormalization())

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

model.add(Dropout(0.5))

#3rd convolution layer

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

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

# model.add(BatchNormalization())

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

model.add(Flatten())

#fully connected neural networks

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\_labels, activation='softmax'))

# model.summary()

#Compliling the model

model.compile(loss=categorical\_crossentropy,

              optimizer=Adam(),

              metrics=['accuracy'])

#Training the model

model.fit(X\_train, train\_y,

          batch\_size=batch\_size,

          epochs=epochs,

          verbose=1,

          validation\_data=(X\_test, test\_y),

          shuffle=True)

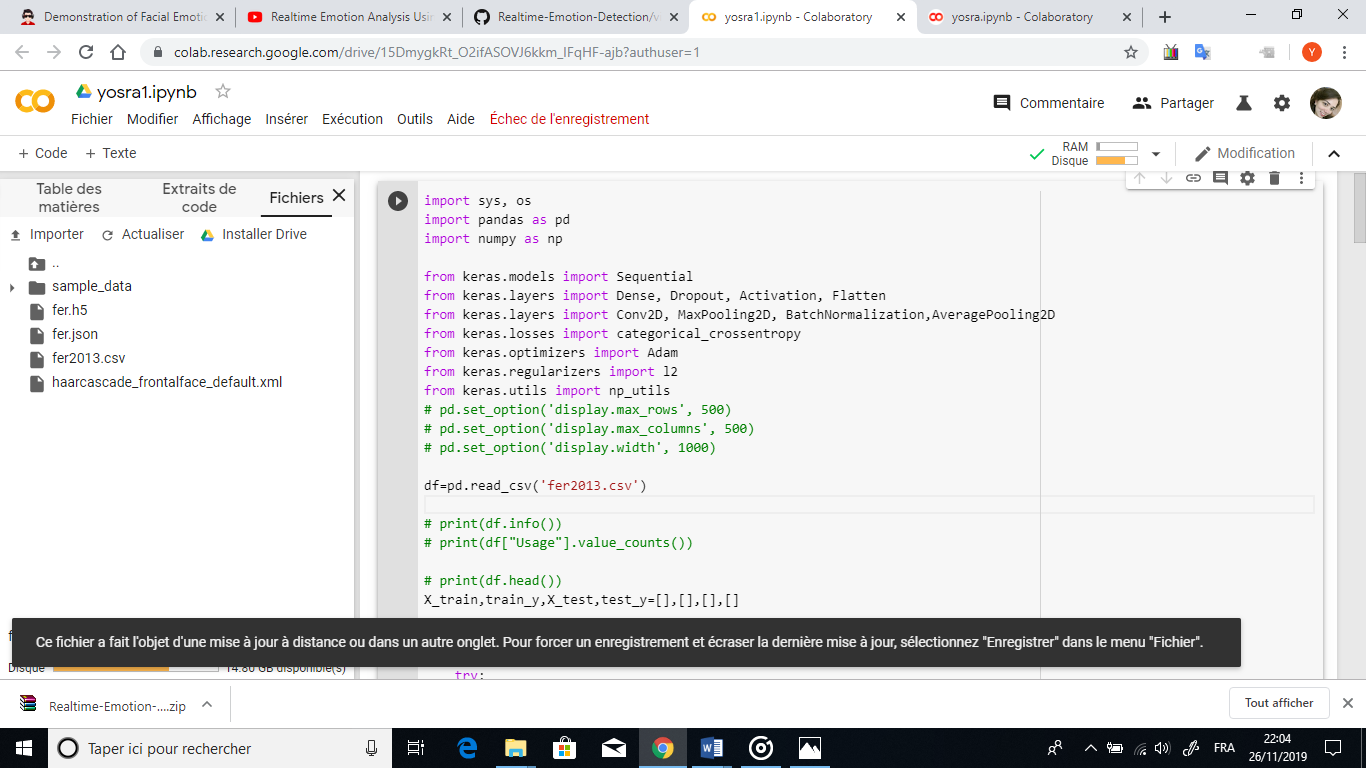
#Saving the  model to  use it later on

fer\_json = model.to\_json()

with open("fer.json", "w") as json\_file:

    json\_file.write(fer\_json)

model.save\_weights("fer.h5")



Using TensorFlow backend.

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.  
We recommend you [upgrade](https://www.tensorflow.org/guide/migrate) now or ensure your notebook will continue to use TensorFlow 1.x via the %tensorflow\_version 1.x magic: [more info](https://colab.research.google.com/notebooks/tensorflow_version.ipynb).

error occured at index :12880 and row:emotion 2

pixels 251 251 253 253 253 254 254 254 254 254 254 25...

Usage NaN

Name: 12880, dtype: object

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:66: The name tf.get\_default\_graph is deprecated. Please use tf.compat.v1.get\_default\_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:541: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:4432: The name tf.random\_uniform is deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:4267: The name tf.nn.max\_pool is deprecated. Please use tf.nn.max\_pool2d instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:148: The name tf.placeholder\_with\_default is deprecated. Please use tf.compat.v1.placeholder\_with\_default instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:3733: calling dropout (from tensorflow.python.ops.nn\_ops) with keep\_prob is deprecated and will be removed in a future version.

Instructions for updating:

Please use `rate` instead of `keep\_prob`. Rate should be set to `rate = 1 - keep\_prob`.

/usr/local/lib/python3.6/dist-packages/numpy/core/fromnumeric.py:3257: RuntimeWarning: Mean of empty slice.

out=out, \*\*kwargs)

/usr/local/lib/python3.6/dist-packages/numpy/core/\_methods.py:161: RuntimeWarning: invalid value encountered in true\_divide

ret = ret.dtype.type(ret / rcount)

/usr/local/lib/python3.6/dist-packages/numpy/core/\_methods.py:217: RuntimeWarning: Degrees of freedom <= 0 for slice

keepdims=keepdims)

/usr/local/lib/python3.6/dist-packages/numpy/core/\_methods.py:186: RuntimeWarning: invalid value encountered in true\_divide

arrmean, rcount, out=arrmean, casting='unsafe', subok=False)

/usr/local/lib/python3.6/dist-packages/numpy/core/\_methods.py:209: RuntimeWarning: invalid value encountered in true\_divide

ret = ret.dtype.type(ret / rcount)

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:793: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:3576: The name tf.log is deprecated. Please use tf.math.log instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow\_core/python/ops/math\_grad.py:1424: where (from tensorflow.python.ops.array\_ops) is deprecated and will be removed in a future version.

Instructions for updating:

Use tf.where in 2.0, which has the same broadcast rule as np.where

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:1033: The name tf.assign\_add is deprecated. Please use tf.compat.v1.assign\_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:1020: The name tf.assign is deprecated. Please use tf.compat.v1.assign instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:3005: The name tf.Session is deprecated. Please use tf.compat.v1.Session instead.

Train on 12880 samples, validate on 0 samples

Epoch 1/30

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:190: The name tf.get\_default\_session is deprecated. Please use tf.compat.v1.get\_default\_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:197: The name tf.ConfigProto is deprecated. Please use tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:207: The name tf.global\_variables is deprecated. Please use tf.compat.v1.global\_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:216: The name tf.is\_variable\_initialized is deprecated. Please use tf.compat.v1.is\_variable\_initialized instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:223: The name tf.variables\_initializer is deprecated. Please use tf.compat.v1.variables\_initializer instead.

12880/12880 [==============================] - 239s 19ms/step - loss: 1.7962 - acc: 0.2488

Epoch 2/30

12880/12880 [==============================] - 238s 19ms/step - loss: 1.6809 - acc: 0.3238

Epoch 3/30

12880/12880 [==============================] - 237s 18ms/step - loss: 1.5624 - acc: 0.3804

Epoch 4/30

12880/12880 [==============================] - 238s 18ms/step - loss: 1.4836 - acc: 0.4217

Epoch 5/30

12880/12880 [==============================] - 239s 19ms/step - loss: 1.4268 - acc: 0.4391

Epoch 6/30

12880/12880 [==============================] - 240s 19ms/step - loss: 1.3745 - acc: 0.4707

Epoch 7/30

12880/12880 [==============================] - 238s 19ms/step - loss: 1.3360 - acc: 0.4835

Epoch 8/30

12880/12880 [==============================] - 238s 18ms/step - loss: 1.2922 - acc: 0.5040

Epoch 9/30

12880/12880 [==============================] - 238s 19ms/step - loss: 1.2636 - acc: 0.5085

Epoch 10/30

12880/12880 [==============================] - 238s 19ms/step - loss: 1.2352 - acc: 0.5241

Epoch 11/30

12880/12880 [==============================] - 238s 19ms/step - loss: 1.1992 - acc: 0.5328

Epoch 12/30

12880/12880 [==============================] - 237s 18ms/step - loss: 1.1740 - acc: 0.5458

Epoch 13/30

12880/12880 [==============================] - 238s 18ms/step - loss: 1.1533 - acc: 0.5511

Epoch 14/30

12880/12880 [==============================] - 239s 19ms/step - loss: 1.1227 - acc: 0.5646

Epoch 15/30

12880/12880 [==============================] - 237s 18ms/step - loss: 1.0948 - acc: 0.5775

Epoch 16/30

12880/12880 [==============================] - 238s 18ms/step - loss: 1.0530 - acc: 0.6004

Epoch 17/30

12880/12880 [==============================] - 240s 19ms/step - loss: 1.0257 - acc: 0.6072

Epoch 18/30

12880/12880 [==============================] - 240s 19ms/step - loss: 0.9906 - acc: 0.6194

Epoch 19/30

12880/12880 [==============================] - 239s 19ms/step - loss: 0.9800 - acc: 0.6227

Epoch 20/30

12880/12880 [==============================] - 241s 19ms/step - loss: 0.9364 - acc: 0.6411

Epoch 21/30

12880/12880 [==============================] - 241s 19ms/step - loss: 0.9161 - acc: 0.6501

Epoch 22/30

12880/12880 [==============================] - 241s 19ms/step - loss: 0.8852 - acc: 0.6605

Epoch 23/30

12880/12880 [==============================] - 241s 19ms/step - loss: 0.8691 - acc: 0.6718

Epoch 24/30

12880/12880 [==============================] - 241s 19ms/step - loss: 0.8481 - acc: 0.6832

Epoch 25/30

12880/12880 [==============================] - 240s 19ms/step - loss: 0.8225 - acc: 0.6884

Epoch 26/30

12880/12880 [==============================] - 240s 19ms/step - loss: 0.8006 - acc: 0.6987

Epoch 27/30

12880/12880 [==============================] - 239s 19ms/step - loss: 0.7772 - acc: 0.7089

Epoch 28/30

12880/12880 [==============================] - 240s 19ms/step - loss: 0.7434 - acc: 0.7227

Epoch 29/30

12880/12880 [==============================] - 238s 18ms/step - loss: 0.7267 - acc: 0.7294

Epoch 30/30

12880/12880 [==============================] - 238s 19ms/step - loss: 0.7103 - acc: 0.7306

<https://www.youtube.com/watch?v=DtBu1u5aBsc>

<https://github.com/neha01/Realtime-Emotion-Detection>

    import os

    import cv2

    import numpy as np

    from keras.models import model\_from\_json

    from keras.preprocessing import image

#load model

    model = model\_from\_json(open("fer.json", "r").read())

#load weights

    model.load\_weights('fer.h5')

    face\_haar\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

    test\_img = cv2.imread('/content/IMG13382\_resultat.jpg')

    gray\_img = cv2.cvtColor(test\_img, cv2.COLOR\_BGR2GRAY)

    faces\_detected = face\_haar\_cascade.detectMultiScale(gray\_img, 1.32, 5)

    for (x,y,w,h) in faces\_detected:

        cv2.rectangle(test\_img,(x,y),(x+w,y+h),(255,0,0),thickness=7)

        roi\_gray=gray\_img[y:y+w,x:x+h]#cropping region of interest i.e. face area from  image

        roi\_gray=cv2.resize(roi\_gray,(48,48))

        img\_pixels = image.img\_to\_array(roi\_gray)

        img\_pixels = np.expand\_dims(img\_pixels, axis = 0)

        img\_pixels /= 255

        predictions = model.predict(img\_pixels)

        #find max indexed array

        max\_index = np.argmax(predictions[0])

        emotions = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')

        predicted\_emotion = emotions[max\_index]

        cv2.putText(test\_img, predicted\_emotion, (int(x), int(y)), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0,0,255), 2)

    resized\_img = cv2.resize(test\_img, (1000, 700))

    import matplotlib.pyplot as plt

    plt.imshow(resized\_img)

    plt.show()

cv2.destroyAllWindows

