

Import des bibliothèques

Entrée []:

```
import pandas as pd
import numpy as np
import cv2
import tensorflow as tf
import os

from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import Dense, Flatten, Conv2D
from tensorflow.keras import Model
from PIL import Image

from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras import Sequential
from tensorflow.keras.optimizers import Adam
from keras.layers.normalization import BatchNormalization
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.layers import Dropout

from sklearn.model_selection import train_test_split
```

Entrée []:

```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))
```

Importation des données

import angry

Entrée []:

```
header = [str(i) for i in range(2304)]
```

Entrée []:

```
dfAngryTrain = pd.DataFrame(np.array([0 for i in range(2304)]))
dfAngryTrain = dfAngryTrain.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/angry'+element+'.png')
    dfAngryTrain = pd.concat([dfAngryTrain, pd.DataFrame(np.array(img.getdata()).T)])
```

Entrée []:

```
dfAngryTest = pd.DataFrame(np.array([0 for i in range(2304)]))
dfAngryTest = dfAngryTest.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/angry'+element+'.png')
    dfAngryTest = pd.concat([dfAngryTest, pd.DataFrame(np.array(img.getdata()).T)])
```

Entrée []:

```
dfAngryTrain["target"] = 0
dfAngryTest["target"] = 0
```

Entrée []:

```
dfAngryTrain = dfAngryTrain.reset_index()
dfAngryTrain
```

Entrée []:

```
dfAngryTest = dfAngryTest.reset_index()
dfAngryTest
```

import Disgust

Entrée []:

```
dfDisgustTrain = pd.DataFrame(np.array([0 for i in range(2304)]))
dfDisgustTrain = dfDisgustTrain.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/disgust/' + element)
    dfDisgustTrain = pd.concat([dfDisgustTrain, pd.DataFrame(np.array(img.getdata()))])
```

Entrée []:

```
dfDisgustTest = pd.DataFrame(np.array([0 for i in range(2304)]))
dfDisgustTest = dfDisgustTest.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/disgust/' + element)
    dfDisgustTest = pd.concat([dfDisgustTest, pd.DataFrame(np.array(img.getdata()))])
```

Entrée []:

```
dfDisgustTrain["target"] = 1
dfDisgustTest["target"] = 1
```

Entrée []:

```
dfDisgustTrain = dfDisgustTrain.reset_index()
dfDisgustTrain
```

Entrée []:

```
dfDisgustTest = dfDisgustTest.reset_index()
dfDisgustTest
```

import Fear

Entrée []:

```
dfFearTrain = pd.DataFrame(np.array([0 for i in range(2304)]))
dfFearTrain = dfFearTrain.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/fear/' + element)
    dfFearTrain = pd.concat([dfFearTrain, pd.DataFrame(np.array(img.getdata()).T)])
```

Entrée []:

```
dfFearTest = pd.DataFrame(np.array([0 for i in range(2304)]))
dfFearTest = dfFearTest.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/fear/' + element)
    dfFearTest = pd.concat([dfFearTest, pd.DataFrame(np.array(img.getdata()).T)])
```

Entrée []:

```
dfFearTrain["target"] = 2
dfFearTest["target"] = 2
```

Entrée []:

```
dfFearTrain = dfFearTrain.reset_index()
dfFearTrain
```

Entrée []:

```
dfFearTest = dfFearTest.reset_index()
dfFearTest
```

import Happy

Entrée []:

```
dfHappyTrain = pd.DataFrame(np.array([0 for i in range(2304)]))
dfHappyTrain = dfHappyTrain.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/happy/' + element)
    dfHappyTrain = pd.concat([dfHappyTrain, pd.DataFrame(np.array(img.getdata()).T)])
```

Entrée []:

```
dfHappyTest = pd.DataFrame(np.array([0 for i in range(2304)]))
dfHappyTest = dfHappyTest.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/happy/' + element)
    dfHappyTest = pd.concat([dfHappyTest, pd.DataFrame(np.array(img.getdata()).T)])
```

Entrée []:

```
dfHappyTrain["target"] = 3
dfHappyTest["target"] = 3
```

Entrée []:

```
dfHappyTrain = dfHappyTrain.reset_index()  
dfHappyTrain
```

Entrée []:

```
dfHappyTest = dfHappyTest.reset_index()  
dfHappyTest
```

import Sad

Entrée []:

```
dfSadTrain = pd.DataFrame(np.array([0 for i in range(2304)]))  
dfSadTrain = dfSadTrain.T  
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/'):   
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/sad/' + element + '.png')  
    dfSadTrain = pd.concat([dfSadTrain, pd.DataFrame(np.array(img.getdata()).T)])
```

Entrée []:

```
dfSadTest = pd.DataFrame(np.array([0 for i in range(2304)]))  
dfSadTest = dfSadTest.T  
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/'):   
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/sad/' + element + '.png')  
    dfSadTest = pd.concat([dfSadTest, pd.DataFrame(np.array(img.getdata()).T)])
```

Entrée []:

```
dfSadTrain["target"] = 4  
dfSadTest["target"] = 4
```

Entrée []:

```
dfSadTrain = dfSadTrain.reset_index()  
dfSadTrain
```

Entrée []:

```
dfSadTest = dfSadTest.reset_index()  
dfSadTest
```

import Surprise

Entrée []:

```
dfSurpriseTrain = pd.DataFrame(np.array([0 for i in range(2304)]))  
dfSurpriseTrain = dfSurpriseTrain.T  
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/'):   
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/surp/' + element + '.png')  
    dfSurpriseTrain = pd.concat([dfSurpriseTrain, pd.DataFrame(np.array(img.getdata()).T)])
```

Entrée []:

```
dfSurpriseTest = pd.DataFrame(np.array([0 for i in range(2304)]))
dfSurpriseTest = dfSurpriseTest.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/surpr')
    dfSurpriseTest = pd.concat([dfSurpriseTest, pd.DataFrame(np.array(img.getdata()))])
```

Entrée []:

```
dfSurpriseTrain["target"] = 5
dfSurpriseTest["target"] = 5
```

Entrée []:

```
dfSurpriseTrain = dfSurpriseTrain.reset_index()
dfSurpriseTrain
```

Entrée []:

```
dfSurpriseTest = dfSurpriseTest.reset_index()
dfSurpriseTest
```

import Neutral

Entrée []:

```
dfNeutralTrain = pd.DataFrame(np.array([0 for i in range(2304)]))
dfNeutralTrain = dfNeutralTrain.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/train/neut')
    dfNeutralTrain = pd.concat([dfNeutralTrain, pd.DataFrame(np.array(img.getdata()))])
```

Entrée []:

```
dfNeutralTest = pd.DataFrame(np.array([0 for i in range(2304)]))
dfNeutralTest = dfNeutralTest.T
for element in os.listdir('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/'):
    img = Image.open('/Users/fabiodjs/Desktop/ProjetLaure/Dossier/archive/test/neutr')
    dfNeutralTest = pd.concat([dfNeutralTest, pd.DataFrame(np.array(img.getdata()))]).T
```

Entrée []:

```
dfNeutralTrain["target"] = 6
dfNeutralTest["target"] = 6
```

Entrée []:

```
dfNeutralTrain = dfNeutralTrain.reset_index()
dfNeutralTrain
```

Entrée []:

```
dfNeutralTest = dfNeutralTest.reset_index()
dfNeutralTest
```

création du tableau global

Entrée []:

```
dfFinalTrain = pd.concat([dfAngryTrain,dfDisgustTrain,dfFearTrain,dfHappyTrain,dfSadTrain])
dfFinalTrain = dfFinalTrain.reset_index()
dfFinalTrain
```

Entrée []:

```
dfFinalTest = pd.concat([dfAngryTest,dfDisgustTest,dfFearTest,dfHappyTest,dfSadTest])
dfFinalTest = dfFinalTest.reset_index()
dfFinalTest
```

Création des CSV

Entrée []:

```
dfFinalTrain.to_csv("datasetImagesTrain.csv",index=False)
```

Entrée []:

```
dfFinalTest.to_csv("datasetImagesTest.csv",index=False)
```

Entrée []:

```
del dfFinalTest["level_0"]
del dfFinalTest["index"]

del dfFinalTrain["level_0"]
del dfFinalTrain["index"]

dfFinalTrain.to_csv("datasetImagesTrain.csv", index = False)
dfFinalTest.to_csv("datasetImagesTest.csv", index = False)
```

Lecture fichier CSV

Entrée []:

```
dfFinalTrain = pd.read_csv('./datasetImagesTrain.csv')
```

Entrée []:

```
dfFinalTest = pd.read_csv('./datasetImagesTest.csv')
```

Création des variables nécessaires au fonctionnement du model

Entrée []:

```
trainX = dfFinalTrain.loc[:, dfFinalTrain.columns != 'target']
trainX = trainX/255.0
trainY = dfFinalTrain['target']

trainX = trainX.values.reshape(len(trainX),48,48,1)
trainY = tf.keras.utils.to_categorical(trainY,7)
```

Entrée []:

```
testX = dfFinalTest.loc[:, dfFinalTest.columns != 'target']
testX = testX/255.0
testY = dfFinalTest['target']

testX = testX.values.reshape(len(testX),48,48,1)
testY = tf.keras.utils.to_categorical(testY, )
testY = np.argmax(testY , axis=1)
```

Création du model

Entrée []:

```
nombreDeClasse = 7
img_height = 48
img_width = 48
img_depth = 1

with tf.device('/device:GPU:0'):
    model = Sequential()
    model.add(Conv2D(filters=64,kernel_size=(3,3),input_shape=(img_height, img_width,
    model.add(BatchNormalization())
    model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu',padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.4))
    model.add(Conv2D(filters=128,kernel_size=(3,3),activation='relu',padding='same'))
    model.add(BatchNormalization())
    model.add(Conv2D(filters=128,kernel_size=(3,3),activation='relu',padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.4))
    model.add(Conv2D(filters=256,kernel_size=(3,3),activation='relu',padding='same'))
    model.add(BatchNormalization())
    model.add(Conv2D(filters=256,kernel_size=(3,3),padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2), name='maxpool2d_3'))
    model.add(Dropout(0.5))
    model.add(Flatten())
    model.add(Dense(128,activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.6))
    model.add(Dense(7,activation='softmax'))
    model.compile(loss='categorical_crossentropy',optimizer=keras.optimizers.Adam(learn
    model.summary()
```

Entrée []:

```
model.fit(trainX,trainY,epochs=100)
```

Sauvegarde du model en local

Entrée []:

```
model.save("myModel.h5")
```

Importation du model sur le notebook

Entrée []:

```
model = keras.models.load_model("model63acc.h5")
```

Résultats du model sur le test set

Entrée []:

```
from sklearn.metrics import classification_report

pred = model.predict(testX)
pred = np.argmax(pred,axis=1)

print('Classification Report')
print(classification_report(testY, pred, target_names = ["angry" ,"disgust" ,"fear"])
```

Entrée []: