Code of the training of the model

from fastai.vision.all import \*

p2 = "/kaggle/input/fer2013plus"

path = Path(p2)

import tensorflow as tf

from tensorflow import keras

dls = DataBlock(

blocks=(ImageBlock, CategoryBlock),

get\_items=get\_image\_files,

splitter=GrandparentSplitter(train\_name='train', valid\_name='test'),

get\_y=parent\_label,

item\_tfms=[Resize(48, method='squish')]

).dataloaders(path, bs=32)

dls.show\_batch(max\_n=6)

learner = vision\_learner(dls, resnet50, metrics=[accuracy])

learner.fine\_tune(15)

learner.export('model50\_15.pkl')

Code for the application with the model

import matplotlib.pyplot as plt

from fastai.vision.all import \*

#import tensorflow as tf

#from tensorflow import keras

#learn = load\_learner('/kaggle/input/model1/modelv.pkl')

import cv2

p1 = "C:/Users/MSUser/.spyder-py3/Dip/M/1.jpg"

import pathlib

temp = pathlib.PosixPath

pathlib.PosixPath = pathlib.WindowsPath

model = load\_learner('C:/Users/MSUser/Downloads/model50.pkl')

#pathlib.PosixPath = temp

print("imported")

face\_classifier = cv2.CascadeClassifier(

cv2.data.haarcascades + "haarcascade\_frontalface\_default.xml"

)

a= 1

#pas = 1

vs = 1

#video = cv2.VideoCapture('C:Users/MSUser/Downloads/sample/video.mp4')

cap = cv2.VideoCapture(0)

while a <=150:

ret, frame = cap.read()

#print(ret)

if not ret:

break

if vs%2==1 :

face = face\_classifier.detectMultiScale(

frame, scaleFactor=1.1, minNeighbors=5, minSize=(40, 40)

)

for (x, y, w, h) in face:

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 4)

roi = frame[y - 15:y+w +15 , x - 15:h +x +15]

vs = vs +1

cv2.putText(frame,str(a), (x,y+200), cv2.FONT\_HERSHEY\_SIMPLEX, 1, 200)

if vs%2==1 :

pr = model.predict(roi)

if pr[0]=="neutral" :

cv2.putText(frame,str(pr[0]), (x,y +200), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0,255,0))

else :

cv2.putText(frame,str(pr[0]), (x,y +200), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0,0,255))

#cv2.putText(frame,str(pr[0]), (x,y +100), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0,255,0))

#cv2.putText(frame,str(pr[0]), (x,y +200), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0,255,0))

cv2.imshow("frame",frame)

#plt.figure(figsize=(2,1))

#plt.imshow(roi)

#plt.axis('off')

key = cv2.waitKey(1)

if key == 27:

break

a = a+1

cap.release()

cv2.destroyAllWindows()