#Image test

import cv2

import numpy as np

from keras.models import load\_model

import sys

faceCascade = cv2.CascadeClassifier('haarcascade\_frontalface\_alt2.xml')

model = load\_model('keras\_model/model\_5-49-0.62.hdf5')

def test\_image(addr):

target = ['angry','disgust','fear','happy','sad','surprise','neutral']

font = cv2.FONT\_HERSHEY\_SIMPLEX

im = cv2.imread(addr)

gray = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

faces = faceCascade.detectMultiScale(gray,scaleFactor=1.1)

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

cv2.rectangle(im, (x, y), (x+w, y+h), (0, 255, 0), 2,5)

face\_crop = im[y:y+h,x:x+w]

face\_crop = cv2.resize(face\_crop,(48,48))

face\_crop = cv2.cvtColor(face\_crop, cv2.COLOR\_BGR2GRAY)

face\_crop = face\_crop.astype('float32')/255

face\_crop = np.asarray(face\_crop)

face\_crop = face\_crop.reshape(1, 1,face\_crop.shape[0],face\_crop.shape[1])

result = target[np.argmax(model.predict(face\_crop))]

cv2.putText(im,result,(x,y), font, 1, (200,0,0), 3, cv2.LINE\_AA)

cv2.imshow('result', im)

cv2.imwrite('result.jpg',im)

cv2.waitKey(0)

if \_\_name\_\_=='\_\_main\_\_':

image\_addres = sys.argv[1]

test\_image(image\_addres)

#############################################################################

#Actual code or testing code

import cv2

import numpy as np

from keras.models import load\_model

faceCascade = cv2.CascadeClassifier('haarcascade\_frontalface\_alt2.xml')

video\_capture = cv2.VideoCapture(0)

model = load\_model('keras\_model/model\_5-49-0.62.hdf5')

target = ['angry','disgust','fear','happy','sad','surprise','neutral']

font = cv2.FONT\_HERSHEY\_SIMPLEX

while True:

# Capture frame-by-frame

ret, frame = video\_capture.read()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = faceCascade.detectMultiScale(gray,scaleFactor=1.1)

# Draw a rectangle around the faces

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

cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2,5)

face\_crop = frame[y:y+h,x:x+w]

face\_crop = cv2.resize(face\_crop,(48,48))

face\_crop = cv2.cvtColor(face\_crop, cv2.COLOR\_BGR2GRAY)

face\_crop = face\_crop.astype('float32')/255

face\_crop = np.asarray(face\_crop)

face\_crop = face\_crop.reshape(1, 1,face\_crop.shape[0],face\_crop.shape[1])

result = target[np.argmax(model.predict(face\_crop))]

cv2.putText(frame,result,(x,y), font, 1, (200,0,0), 3, cv2.LINE\_AA)

# Display the resulting frame

cv2.imshow('Video', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# When everything is done, release the capture

video\_capture.release()

cv2.destroyAllWindows()