駕駛視線偵測

一、參考來源:

<https://www.twblogs.net/a/5ba1ab302b71771a4da99579>

1. 實驗步驟

(1)請先複製以下程式碼

import cv2

import numpy as np

import dlib

import time

import math

import pygame

detector = dlib.get\_frontal\_face\_detector()

predictor = dlib.shape\_predictor(r"C:\Users\user\Downloads\shape\_predictor\_68\_face\_landmarks.dat")

POINTS\_NUM\_LANDMARK = 68

# 獲取最大的人臉

def \_largest\_face(dets):

if len(dets) == 1:

return 0

face\_areas = [ (det.right()-det.left())\*(det.bottom()-det.top()) for det in dets]

largest\_area = face\_areas[0]

largest\_index = 0

for index in range(1, len(dets)):

if face\_areas[index] > largest\_area :

largest\_index = index

largest\_area = face\_areas[index]

print("largest\_face index is {} in {} faces".format(largest\_index, len(dets)))

return largest\_index

# 從dlib的檢測結果抽取姿態估計需要的點座標

def get\_image\_points\_from\_landmark\_shape(landmark\_shape):

if landmark\_shape.num\_parts != POINTS\_NUM\_LANDMARK:

print("ERROR:landmark\_shape.num\_parts-{}".format(landmark\_shape.num\_parts))

return -1, None

#2D image points. If you change the image, you need to change vector

image\_points = np.array([

(landmark\_shape.part(30).x, landmark\_shape.part(30).y), # Nose tip

(landmark\_shape.part(8).x, landmark\_shape.part(8).y), # Chin

(landmark\_shape.part(36).x, landmark\_shape.part(36).y), # Left eye left corner

(landmark\_shape.part(45).x, landmark\_shape.part(45).y), # Right eye right corne

(landmark\_shape.part(48).x, landmark\_shape.part(48).y), # Left Mouth corner

(landmark\_shape.part(54).x, landmark\_shape.part(54).y) # Right mouth corner

], dtype="double")

return 0, image\_points

# 用dlib檢測關鍵點，返回姿態估計需要的幾個點座標

def get\_image\_points(img):

#gray = cv2.cvtColor( img, cv2.COLOR\_BGR2GRAY ) # 圖片調整爲灰色

dets = detector( img, 0 )

if 0 == len( dets ):

print( "ERROR: found no face" )

return -1, None

largest\_index = \_largest\_face(dets)

face\_rectangle = dets[largest\_index]

landmark\_shape = predictor(img, face\_rectangle)

return get\_image\_points\_from\_landmark\_shape(landmark\_shape)

# 獲取翻轉向量和平移向量

def get\_pose\_estimation(img\_size, image\_points ):

# 3D model points.

model\_points = np.array([

(0.0, 0.0, 0.0), # Nose tip

(0.0, -330.0, -65.0), # Chin

(-225.0, 170.0, -135.0), # Left eye left corner

(225.0, 170.0, -135.0), # Right eye right corne

(-150.0, -150.0, -125.0), # Left Mouth corner

(150.0, -150.0, -125.0) # Right mouth corner

])

# Camera internals

focal\_length = img\_size[1]

center = (img\_size[1]/2, img\_size[0]/2)

camera\_matrix = np.array(

[[focal\_length, 0, center[0]],

[0, focal\_length, center[1]],

[0, 0, 1]], dtype = "double"

)

print("Camera Matrix :{}".format(camera\_matrix))

dist\_coeffs = np.zeros((4,1)) # Assuming no lens distortion

(success, rotation\_vector, translation\_vector) = cv2.solvePnP(model\_points, image\_points, camera\_matrix, dist\_coeffs, flags=cv2.SOLVEPNP\_ITERATIVE )

print("Rotation Vector:\n {}".format(rotation\_vector))

print("Translation Vector:\n {}".format(translation\_vector))

return success, rotation\_vector, translation\_vector, camera\_matrix, dist\_coeffs

# 從翻轉向量轉換爲歐拉角

def get\_euler\_angle(rotation\_vector):

# calculate rotation angles

theta = cv2.norm(rotation\_vector, cv2.NORM\_L2)

# transformed to quaterniond

w = math.cos(theta / 2)

x = math.sin(theta / 2)\*rotation\_vector[0][0] / theta

y = math.sin(theta / 2)\*rotation\_vector[1][0] / theta

z = math.sin(theta / 2)\*rotation\_vector[2][0] / theta

ysqr = y \* y

# pitch (x-axis rotation)

t0 = 2.0 \* (w \* x + y \* z)

t1 = 1.0 - 2.0 \* (x \* x + ysqr)

print('t0:{}, t1:{}'.format(t0, t1))

pitch = math.atan2(t0, t1)

# yaw (y-axis rotation)

t2 = 2.0 \* (w \* y - z \* x)

if t2 > 1.0:

t2 = 1.0

if t2 < -1.0:

t2 = -1.0

yaw = math.asin(t2)

# roll (z-axis rotation)

t3 = 2.0 \* (w \* z + x \* y)

t4 = 1.0 - 2.0 \* (ysqr + z \* z)

roll = math.atan2(t3, t4)

print('pitch:{}, yaw:{}, roll:{}'.format(pitch, yaw, roll))

# 單位轉換：將弧度轉換爲度

Y = int((pitch/math.pi)\*180)

X = int((yaw/math.pi)\*180)

Z = int((roll/math.pi)\*180)

return 0, Y, X, Z

def get\_pose\_estimation\_in\_euler\_angle(landmark\_shape, im\_szie):

try:

ret, image\_points =

get\_image\_points\_from\_landmark\_shape(landmark\_shape)

if ret != 0:

print('get\_image\_points failed')

return -1, None, None, None

ret, rotation\_vector, translation\_vector, camera\_matrix, dist\_coeffs = get\_pose\_estimation(im\_szie, image\_points)

if ret != True:

print('get\_pose\_estimation failed')

return -1, None, None, None

ret, pitch, yaw, roll = get\_euler\_angle(rotation\_vector)

if ret != 0:

print('get\_euler\_angle failed')

return -1, None, None, None

euler\_angle\_str = 'Y:{}, X:{}, Z:{}'.format(pitch, yaw, roll)

print(euler\_angle\_str)

return 0, pitch, yaw, roll

except Exception as e:

print('get\_pose\_estimation\_in\_euler\_angle exception:{}'.format(e))

return -1, None, None, None

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

cap = cv2.VideoCapture(0)

while (cap.isOpened()):

start\_time = time.time()

# Read Image

ret, im = cap.read()

if ret != True:

print('read frame failed')

continue

size = im.shape

ret, image\_points = get\_image\_points(im)

if ret != 0:

print('get\_image\_points failed')

continue

ret, rotation\_vector, translation\_vector, camera\_matrix, dist\_coeffs = get\_pose\_estimation(size, image\_points)

if ret != True:

print('get\_pose\_estimation failed')

continue

used\_time = time.time() - start\_time

print("used\_time:{} sec".format(round(used\_time, 3)))

ret, pitch, yaw, roll = get\_euler\_angle(rotation\_vector)

euler\_angle\_str = 'Y:{}, X:{}, Z:{}'.format(pitch, yaw, roll)

print(euler\_angle\_str)

if yaw>=30 or yaw<=-30:

cv2.putText(im, "care full!", (200, 30),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 0, 255), 2)

pygame.mixer.init()

pygame.mixer.music.load(r"C:\Users\user\Downloads\bbb.wav")

pygame.mixer.music.play()

# Project a 3D point (0, 0, 1000.0) onto the image plane.

# We use this to draw a line sticking out of the nose

(nose\_end\_point2D, jacobian) = cv2.projectPoints(np.array([(0.0, 0.0, 1000.0)]), rotation\_vector, translation\_vector, camera\_matrix, dist\_coeffs)

for p in image\_points:

cv2.circle(im, (int(p[0]), int(p[1])), 3, (0,0,255), -1)

p1 = ( int(image\_points[0][0]), int(image\_points[0][1]))

p2 = ( int(nose\_end\_point2D[0][0][0]), int(nose\_end\_point2D[0][0][1]))

cv2.line(im, p1, p2, (255,0,0), 2)

# Display image

#cv2.putText( im, str(rotation\_vector), (0, 100), cv2.FONT\_HERSHEY\_PLAIN, 1, (0, 0, 255), 1 )

cv2.putText( im, euler\_angle\_str, (0, 120), cv2.FONT\_HERSHEY\_PLAIN, 1, (0, 0, 255), 1 )

cv2.imshow("Output", im)

cv2.waitKey(1)

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

cv2.destroyAllWindows()

camera.release()

break

執行結果:



