人臉辨識

一、OpenCV 人臉分類器

01. 影像測試

A. 匯入模塊函數

【01】

```
import cv2
import dlib
import matplotlib.pyplot as plt
import time
```

B. 使用 OpenCV 訓練完成的人臉辨識 Haar 分類模型:

- 回傳的人臉位置為左上角的 x, y 與寬高 w, h
- 回傳的格式為陣列（Array）的型式

【02】

```
img = cv2.imread("Face_test.jpg")
img = cv2.resize(img, None, fx=1.5, fy=1.5)

cv_detector = cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_alt2.xml')

start = time.time()
cv_faces = cv_detector.detectMultiScale(img,scaleFactor=1.2,minNeighbors=3,minSize=(15, 15),flags = cv2.CASCADE_SCALE_IMAGE)
end = time.time()
print("time:%0.3f" %(end - start))

for i in range(len(cv_faces)):
    x = cv_faces[i][0]
    y = cv_faces[i][1]
    w = cv_faces[i][2]
    h = cv_faces[i][3]

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

plt.figure(figsize=(15,10))
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
```

1. time.time(): 返回當前時間的時間點

02. 攝影機測試

A．使用 OpenCV 訓練完成的人臉辨識 Haar 分類模型：

【03】

```
VIDEO_IN = cv2.VideoCapture(0)
while True:
    hasFrame, img = VIDEO_IN.read()
    img = cv2.resize(img, None, fx=0.6, fy=0.6)
    cv_faces = cv_detector.detectMultiScale(img,scaleFactor=1.2,minNeighbors=3,minSize=(15, 15),flags = cv2.CASCADE_SCALE_IMAGE)

    for i in range(len(cv_faces)):
        x = cv_faces[i][0]
        y = cv_faces[i][1]
        w = cv_faces[i][2]
        h = cv_faces[i][3]

    cv2.rectangle(img, (x, y), (x+w, y+h), (0, 0, 255), 4)
    cv2.imshow("Frame", img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
VIDEO_IN.release()
cv2.destroyAllWindows()
```

二、Dlib 人臉分類器

01. 安裝 Dlib

A. 請使用 pip 安裝下列六個套件：

numpy, scipy, matplotlib, scikit-learn, jupyter, opencv-python

B. 再行安裝 Dlib，有兩種安裝方式：

- pip install dlib (不建議)
- Dlib 官網下載包 https://pypi.org/simple/dlib/ (下載 dlib-19.8.1-cp36-cp36m-win_amd64.whl)

02. 影像測試

A. 使用 Dlib 訓練完成的人臉辨識分類模型：

- 回傳的人臉位置為左上角的 x1, y1 與右下角的 x2, y2
- 回傳的格式為 dlib.rectangle 的型式，若要提取其值則是使用 left、top、right 與 bottom

【04】

```
img = cv2.imread("Face_test.jpg")
img = cv2.resize(img, None, fx=1.5, fy=1.5)
dlib_detector = dlib.get_frontal_face_detector()
```

```
start = time.time()
dlib_faces = dlib_detector(img)
end = time.time()
print("time:%0.3f" %(end - start))

for i in range(len(dlib_faces)):
    x1 = dlib_faces[i].left()
    y1 = dlib_faces[i].top()
    x2 = dlib_faces[i].right()
    y2 = dlib_faces[i].bottom()

    cv2.rectangle(img, (x1, y1), (x2, y2), (0, 0, 255), 4)

plt.figure(figsize=(15,10))
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
```

03. 攝影機測試

A．使用 Dlib 訓練完成的人臉辨識分類模型：

【05】

```
VIDEO_IN = cv2.VideoCapture(0)
while True:
    hasFrame, img = VIDEO_IN.read()
    img = cv2.resize(img, None, fx=0.6, fy=0.6)
    dlib_faces = dlib_detector(img)

    for i in range(len(dlib_faces)):
        x1 = dlib_faces[i].left()
        y1 = dlib_faces[i].top()
        x2 = dlib_faces[i].right()
        y2 = dlib_faces[i].bottom()

    cv2.rectangle(img, (x1, y1), (x2, y2), (0, 0, 255), 4)
    cv2.imshow("Frame", img)

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

VIDEO_IN.release()
cv2.destroyAllWindows()
```

表情辨識

一、使用表情辨識模型

01. 人臉與表情辨識分類器及貼圖方法

A. 匯入模塊函數

【01】

```
from tensorflow import keras
import dlib
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

B. 讀取訓練完成的 Keras FER 分類模型與 Dlib 人臉辨識器

【02】

```
model = keras.models.load_model('model.h5')
model.summary()
detector = dlib.get_frontal_face_detector()
```

C. 建立字典與列表，分別為表情的名稱與框的顏色：

- 顏色：0:紅色, 1:黑色, 2:灰色, 3:橘色, 4:藍色, 5:紫色, 6:綠色
- 名稱：0:生氣, 1:厭惡, 2:恐懼, 3:開心, 4:難過, 5:驚訝, 6:中立

【03】

```
expression_dict = {0: "Angry", 1: "Disgust", 2: "Fear", 3: "Happy",
                   4: "Sad", 5: "Surprise", 6: "Neutral"}
colors = [(0,0,255),(0,0,0),(135,138,128),(18,153,255),
          (255,0,0),(240,32,160),(0,255,0)]
```

D. 讀取各表情的貼圖影像：

圖一、自己設計的表情貼圖

【04】

```
img_Angry = cv2.imread("Image_fer/00_Angry_Second.png")
img_Disgust = cv2.imread("Image_fer/01_Disgust_Second.png")
img_Fear = cv2.imread("Image_fer/02_Fear_Second.png")
img_Happy = cv2.imread("Image_fer/03_Happy_Second.png")
img_Sad = cv2.imread("Image_fer/04_Sad_Second.png")
img_Surprise = cv2.imread("Image_fer/05_Surprise_Second.png")
img_Neutral = cv2.imread("Image_fer/06_Neutral_Second.png")
```

E. 建立一 detect_faces 函數，將人臉位置 dlib.rectangle 的格式轉為列表，並把整個影像轉為灰階影像:

【05】

```
def detect_faces(frame):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    face = detector(frame)
    faces = []
    for i in range(len(face)):
        x = face[i].left()
        y = face[i].top()
        w = face[i].right()-face[i].left()
        h = face[i].bottom()-face[i].top()
        if x<0:
            x = 0
        if y<0:
            y = 0
        faces.append((x, y, w, h))
    return gray, faces
```

F. 建立 face_check 函數

【06】

```
def face_check(frame):
    gray, faces = detect_faces(frame)
    for (x, y, w, h) in faces:
        roi_gray = gray[y:y + h, x:x + w]
        cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray,(48, 48)),-1), 0)
        prediction = model.predict(cropped_img)[0]
        pred_list = prediction.tolist()
        pred_max = max(pred_list)
        index = pred_list.index(pred_max)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 4)
        cv2.putText(frame,"{}: {}".format(expression_dict[index],round(pred_max,3)),
                    (x, y-10),cv2.FONT_HERSHEY_SIMPLEX,0.5,colors[index],2)
        frame = face_post(frame, (x, y), w, expression_dict[index])
    return frame
```

G. 建立 face_post 函數

【07】

```
def face_post(frame, top_left, width, prediction):
    if prediction =="Angry":
        img = img_Angry
    if prediction =="Disgust":
        img = img_Disgust
    if prediction =="Fear":
        img = img_Fear
```

```
    if prediction =="Happy":
        img = img_Happy
    if prediction =="Sad":
        img = img_Sad
    if prediction =="Surprise":
        img = img_Surprise
    if prediction =="Neutral":
        img = img_Neutral
    img_wh = int(width/2)
    img = cv2.resize(img, (img_wh,img_wh))

    img_top_left = (int(top_left[0]-img_wh),int(top_left[1]))
    img_bottom_right = (int(top_left[0]),int(top_left[1]+img_wh))

    if img_top_left[0]>0 and img_top_left[1]>0:
        img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        _, mask = cv2.threshold(img_gray, 25, 255, cv2.THRESH_BINARY_INV)
        area = frame[img_top_left[1]: img_top_left[1]+img_wh, img_top_left[0]: img_top
_left[0]+img_wh]
        area_no = cv2.bitwise_and(area, area, mask = mask)
        final = cv2.add(area_no, img)
        frame[img_top_left[1]: img_top_left[1]+img_wh, img_top_left[0]: img_top_left
[0]+img_wh] = final

    return frame
```

2. and_img = cv2.bitwise_and(img1, img2, mask)：是對二進制數據進行「and」操作，例如：1&1=1，1&0=0，0&1=0，0&0=0。若 mask（x, y）！= 0，and_img（x, y）= img1（x, y）^ img2（x, y），否則 and_img（x, y）保留其原始值，並且 and_img 陣列的所有元素的預設值為 0。其他 bitwise_or、bitwise_xor 與 bitwise_not 等函數也有相同的功能
3. output_array = cv2.add(array1, array2)：兩個陣列像素相加，注意兩個陣列的大小類型必須一樣

02. 影像測試

A. 以影像測試 face_check 函數

【08】

```
img = cv2.imread("Face_test.jpg")
img = cv2.resize(img, None, fx=1.4, fy=1.4)
img = face_check(img)
```

```
plt.figure(figsize=(15,15))
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.show()
```

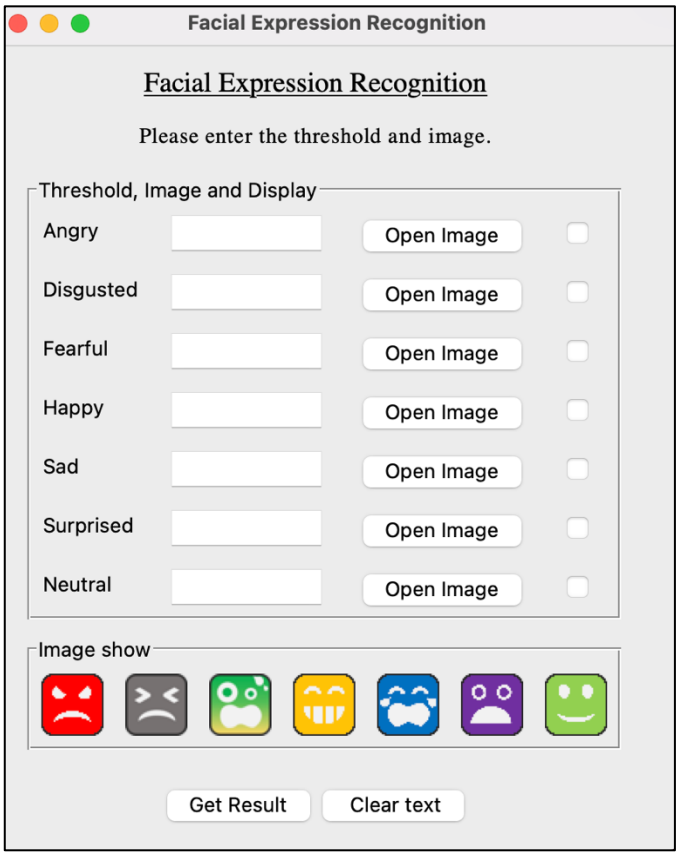
B. 以攝影機測試 face_check 函數

【09】

```
VIDEO_IN = cv2.VideoCapture(0)

while True:
    hasFrame, img = VIDEO_IN.read()
    img = cv2.resize(img, None, fx=0.5, fy=0.5)
    img = face_check(img)
    cv2.imshow("Frame", img)

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

VIDEO_IN.release()
cv2.destroyAllWindows()
```

二、設計表情辨識使用者介面

01. 建立 GUI 介面



圖二、GUI 介面

A. 匯入套件與模型

【10】

```
from tensorflow import keras
import dlib
import cv2
import numpy as np
import tkinter as tk
model = keras.models.load_model('model_FER.h5')
detector = dlib.get_frontal_face_detector()
expression_dict = {0: "Angry", 1: "Disgust", 2: "Fear", 3: "Happy", 4: "Sad", 5: "Surprise", 6: "Neutral"}
colors = [(0,0,255),(0,0,0),(135,138,128),(18,153,255),(255,0,0),(240,32,160),(0,255,0)]
```

B. 建立 GUI 視窗

【11】

```
app = tk.Tk() #建立視窗
app.title('FER') #建立視窗標題
app.geometry('440x520') #建立視窗大小
app.mainloop() #程式開始無限迴圈
```

C. 佈局與放置標籤文字

【12】

```
#建立畫布與位置
frame_title = tk.Frame(app)
frame_title.grid(column=0, row=0, ipadx=5, pady=5)

frame_input = tk.LabelFrame(app, text='Threshold, Image and Display')
frame_input.grid(column=0, row=1, padx=20, sticky=tk.W+tk.N)

frame_img = tk.LabelFrame(app, text='Image show')
frame_img.grid(column=0, row=2, padx=20, sticky=tk.W+tk.N)

frame_output = tk.Frame(app)
frame_output.grid(column=0, row=3)

# 建立標題與位置
label_Title = tk.Label(frame_title,text="Facial Expression Recognition")
label_Title.grid(column=0, row=0, ipadx=5, pady=5)

label_Subtitle = tk.Label(frame_title,text="Please enter the threshold and image.")
label_Subtitle.grid(column=0, row=1, ipadx=5, pady=5)
```

【13】

```
# 範例 1 放置標籤，並顯示佈局
label_input = tk.Label(frame_input,text="Label Input")
label_input.grid(column=0, row=0)

label_img = tk.Label(frame_img,text="Label Img")
label_img.grid(column=0, row=0)

label_output = tk.Label(frame_output,text="Label Output")
label_output.grid(column=0, row=0)
```

4. Var = tk.Frame(x)：為框架控件，在屏幕上顯示一個矩形區域作為容器, x 為放置的視窗
5. Var.grid(column=i, row=j, sticky, padx, pady)：可將框架、標籤等控件(Var)放入指定的位置，如表格方式呈現，將控件放入第 i 欄，第 j 列, sticky 可以為 tk.N,W,S,E 分別是靠上左下右。ipadx, ipady, padx, pady 為內與外的左邊與上面的間隙距離。其他放置位置的方法類似的有 pack(), place()
6. Var = tk.Label(x, text="text" , image=img)：為標籤控件，可以顯示文字與圖片，text 為顯示字串文字，image 為 tk 影像, x 為放置的框架
7. Var = tk.LabelFrame(x, text='text')：為文字虛框的容器控件，常用與複雜的窗口佈局, x 為放置的視窗, text 為顯示字串文字

D. 批量放置標籤控件

➢ 放置單個

【14】

```
# 範例 2-1 單個標籤
label = tk.Label(frame_input, text = 'Angry')
label.grid(column=0, row=0, ipadx=0, pady=0, sticky=tk.W+tk.N)
```

➢ for 迴圈放置多個

【15】

```
# 範例 2-2 多個標籤
label_name = ['Angry','Disgusted','Fearful','Happy','Sad','Surprised','Neutral']
for i in range(len(label_name)):
    label = tk.Label(frame_input,text = label_name[i])
    label.grid(column=0, row=i, ipadx=5, pady=5, sticky=tk.W+tk.N)
```

E. 批量放置輸入欄控件

➢ 放置單個

【16】

```
# 範例 3-1 單個輸入欄
entry = tk.Entry(frame_input, width=10) entry.grid(column=1, row=0,
padx=10, pady=5, sticky=tk.N)
```

➢ for 迴圈放置多個

【17】

```
# 範例 3-2 多個輸入欄
entry_list = []
```

```
for i in range(len(label_name)):
    TkEntry = tk.Entry(frame_input, width=10)
    entry_list.append(TkEntry)
    entry_list[i].grid(column=1, row=i, padx=10, pady=5, sticky=tk.N)
```

F. 批量放置按鈕控件

➢ 放置單個

【18】
```
# 範例 4-1 單個按鈕
button = tk.Button(frame_input, text = 'Open Image', command = None)
button.grid(column=2, row=i, padx=10, pady=5, sticky=tk.N)
```

➢ for 迴圈放置多個

【19】
```
# 範例 4-2 多個按鈕
button_list = []
for i in range(len(label_name)):
    TkButton = tk.Button(frame_input, text = 'Open Image', command = None)
    button_list.append(TkButton)
    button_list[i].grid(column=2, row=i, padx=10, pady=5, sticky=tk.N)
```

G. 批量放置多選勾選框控件

➢ 放置單個

【20】
```
# 範例 4-1 單個多選勾選框
chkValue = tk.BooleanVar()
chkValue.set(False)
chkb = tk.Checkbutton(frame_input, text=None, var=chkValue)
chkb.grid(column=3, row=0, padx=10, pady=5, sticky=tk.W)
```

➢ for 迴圈放置多個

【21】
```
# 範例 4-2 多個多選勾選框
chk_Var_list = []
for i in range(len(label_name)):
    chkValue = tk.BooleanVar()
    chkValue.set(False)
    chk_Var_list.append(chkValue)
    chkb = tk.Checkbutton(frame_input, text=None, var=chk_Var_list[i])
    chkb.grid(column=3, row=i, padx=10, pady=5, sticky=tk.W)
```

H. 批量放置標籤圖片控件

➢ 放置單個

【22】
```
# 範例 5-1 單個圖片
img_path_Angry = "Image_fer/00_Angry_Second.png"
```

```
img = Image.open(img_path_Angry).resize((40,40))
imgtk = ImageTk.PhotoImage(img)
label_img = tk.Label(frame_img, image=imgtk)
label_img.grid(column=i, row=0, ipadx=5, pady=5)
```

➢ for 迴圈放置多個

【23】

```
# 範例 5-2 多個圖片
img_path = ["Image_fer/00_Angry_Second.png",
            "Image_fer/01_Disgust_Second.png",
            "Image_fer/02_Fear_Second.png",
            "Image_fer/03_Happy_Second.png",
            "Image_fer/04_Sad_Second.png",
            "Image_fer/05_Surprise_Second.png",
            "Image_fer/06_Neutral_Second.png"]

imgtk_list, label_img_list = [], []

for i in range(len(label_name)):
    img = Image.open(img_path[i]).resize((40,40))
    imgtk = ImageTk.PhotoImage(img)
    imgtk_list.append(imgtk)
    label_img = tk.Label(frame_img, image=imgtk_list[i])
    label_img_list.append(label_img)
    label_img_list[i].grid(column=i, row=0, ipadx=5, pady=5)
```

H. 批量建立 TK 字串變數

➢ for 迴圈放置多個

【24】

```
# 範例 6 多個 Tk String 變數
imgpath_Var = []
for i in range(len(label_name)):
    var = tk.StringVar()
    imgpath_Var.append(var)
```

I. 將有迴圈的多個控件進行總整理

【25】

```
label_name = ['Angry','Disgusted','Fearful','Happy','Sad','Surprised','Neutral']
img_path = ["Image_fer/00_Angry_Second.png",
            "Image_fer/01_Disgust_Second.png",
            "Image_fer/02_Fear_Second.png",
            "Image_fer/03_Happy_Second.png",
            "Image_fer/04_Sad_Second.png",
            "Image_fer/05_Surprise_Second.png",
```

```
            "Image_fer/06_Neutral_Second.png"]

entry_list, button_list, chk_Var_list, imgtk_list, label_img_list, imgpat
h_Var = [],[],[],[],[],[]

for i in range(len(label_name)):
    label = tk.Label(frame_input,text = label_name[i])
    label.grid(column=0, row=i, ipadx=5, pady=5, sticky=tk.W+tk.N)
    TkEntry = tk.Entry(frame_input, width=10)
    entry_list.append(TkEntry)
    entry_list[i].grid(column=1, row=i, padx=10, pady=5, sticky=tk.N)
    TkButton = tk.Button(frame_input, text = 'Open Image', command = None)
    button_list.append(TkButton)
    button_list[i].grid(column=2, row=i, padx=10, pady=5, sticky=tk.N)
    chkValue = tk.BooleanVar()
    chkValue.set(False)
    chk_Var_list.append(chkValue)
    chkb = tk.Checkbutton(frame_input, text=None, var=chk_Var_list[i])
    chkb.grid(column=3, row=i, padx=10, pady=5, sticky=tk.W)
    img = Image.open(img_path[i]).resize((40,40))
    imgtk = ImageTk.PhotoImage(img)
    imgtk_list.append(imgtk)
    label_img = tk.Label(frame_img, image=imgtk_list[i])
    label_img_list.append(label_img)
    label_img_list[i].grid(column=i, row=0, ipadx=5, pady=5)
    var = tk.StringVar()
    imgpath_Var.append(var)
```

J. 建立觸發開啟攝影機與清空輸入欄的按鈕

【26】

```
# 建立觸發開啟攝影機的按鈕與位置
result_Button=tk.Button(frame_output, text='Get Result', command=open_viedo)
result_Button.grid(column=0, row=0, pady=10)

# 建立清空輸入欄的按鈕與位置
clear_button=tk.Button(frame_output, text="Clear text", command=clear_text)
clear_button.grid(column=1, row=0, pady=10)
```

02. 建立按鈕的觸發事件

A. 更換影像

【27】

```
def open_img_Angry():
    filename = askopenfilename(title='Select file', filetypes=[("all files","
*.*"),("png files","*.png"),("jpg files","*.jpg"),("jpeg files","*.jpeg")])
    imgtk = ImageTk.PhotoImage(Image.open(filename).resize((40,40)))
    label_img_list[0].config(image = imgtk)
    label_img_list[0].image = imgtk
    imgpath_Var[0].set(filename)

def open_img_Disgusted():
    filename = askopenfilename(title='Select file', filetypes=[("all files","
*.*"),("png files","*.png"),("jpg files","*.jpg"),("jpeg files","*.jpeg")])
    imgtk = ImageTk.PhotoImage(Image.open(filename).resize((40,40)))
    label_img_list[1].config(image = imgtk)
    label_img_list[1].image = imgtk
    imgpath_Var[1].set(filename)

def open_img_Fearful():
    filename = askopenfilename(title='Select file', filetypes=[("all files","
*.*"),("png files","*.png"),("jpg files","*.jpg"),("jpeg files","*.jpeg")])
    imgtk = ImageTk.PhotoImage(Image.open(filename).resize((40,40)))
    label_img_list[2].config(image = imgtk)
    label_img_list[2].image = imgtk
    imgpath_Var[2].set(filename)

def open_img_Happy():
    filename = askopenfilename(title='Select file', filetypes=[("all files","
*.*"),("png files","*.png"),("jpg files","*.jpg"),("jpeg files","*.jpeg")])
    imgtk = ImageTk.PhotoImage(Image.open(filename).resize((40,40)))
    label_img_list[3].config(image = imgtk)
    label_img_list[3].image = imgtk
    imgpath_Var[3].set(filename)

def open_img_Sad():
    filename = askopenfilename(title='Select file', filetypes=[("all files","
*.*"),("png files","*.png"),("jpg files","*.jpg"),("jpeg files","*.jpeg")])
    imgtk = ImageTk.PhotoImage(Image.open(filename).resize((40,40)))
    label_img_list[4].config(image = imgtk)
    label_img_list[4].image = imgtk
    imgpath_Var[4].set(filename)
```

```
def open_img_Surprised():
    filename = askopenfilename(title='Select file', filetypes=[("all files","
*.*"),("png files","*.png"),("jpg files","*.jpg"),("jpeg files","*.jpeg")])
    imgtk = ImageTk.PhotoImage(Image.open(filename).resize((40,40)))
    label_img_list[5].config(image = imgtk)
    label_img_list[5].image = imgtk
    imgpath_Var[5].set(filename)

def open_img_Neutral():
    filename = askopenfilename(title='Select file', filetypes=[("all files","
*.*"),("png files","*.png"),("jpg files","*.jpg"),("jpeg files","*.jpeg")])
    imgtk = ImageTk.PhotoImage(Image.open(filename).resize((40,40)))
    label_img_list[6].config(image = imgtk)
    label_img_list[6].image = imgtk
    imgpath_Var[6].set(filename)
```

備註：建立好更換影像的方法後，將程式第【26】中新增:

1. `fun_command=[open_img_Angry,open_img_Disgusted,open_img_Fearful, open_img_Happy, open_img_Sad,open_img_Surprised,open_img_Neutral] # 額外新增`
2. `tk.Button` 中的 `command = None`，None 改為 `fun_command[i]`

B. 一鍵清空輸入欄

【28】

```
def clear_text():
    for i in range(len(label_name)):
        entry_list[i].delete('0', 'end')
```

C. 開啟攝影機

【29】

```
def open_viedo():
    VIDEO_IN = cv2.VideoCapture(0)
    while True:
        hasFrame, img = VIDEO_IN.read()
        img = cv2.resize(img, None, fx=0.5, fy=0.5)
        img = face_check(img)
        cv2.imshow("Frame", img)

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

    VIDEO_IN.release()
    cv2.destroyAllWindows()
```

D. 人臉預測：轉灰階與列表

【30】

```
def detect_faces(frame):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    face = detector(frame)
    faces = []
    for i in range(len(face)):
        x = face[i].left()
        y = face[i].top()
        w = face[i].right()-face[i].left()
        h = face[i].bottom()-face[i].top()
        if x<0:
            x = 0
        if y<0:
            y = 0
        faces.append((x, y, w, h))
    return gray, faces
```

E. 表情預測：前處理，形狀轉換，模型預測，繪製框與文字，貼表情圖

【31】

```
def face_check(frame):
    gray, faces = detect_faces(frame)
    for (x, y, w, h) in faces:
        roi_gray = gray[y:y + h, x:x + w]
        cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray,(48, 48)),-1), 0)
        prediction = model.predict(cropped_img)[0]
        pred_list = prediction.tolist()
        pred_max = max(pred_list)
        index = pred_list.index(pred_max)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 4)
        cv2.putText(frame,"{}: {}".format(expression_dict[index],round(pred_max,3)),(x, y-10),cv2.FONT_HERSHEY_SIMPLEX,0.5,colors[index],2)
        frame = face_post(frame, (x, y), w, expression_dict[index])
    return frame
```

E. 表情預測：前處理，形狀轉換，模型預測，繪製框與文字，貼表情圖

【32】

```
def face_post(frame, top_left, width, prediction):
    if prediction =="Angry":
        if imgpath_Var[0].get()!= "":
            img = cv2.imread(imgpath_Var[0].get())
        else:
            img = cv2.imread(img_path[0])
```

```
    if prediction =="Disgust":
        if imgpath_Var[1].get()!= "":
            img = cv2.imread(imgpath_Var[1].get())
        else:
            img = cv2.imread(img_path[1])
    if prediction =="Fear":
        if imgpath_Var[2].get()!= "":
            img = cv2.imread(imgpath_Var[2].get())
        else:
            img = cv2.imread(img_path[2])
    if prediction =="Happy":
        if imgpath_Var[3].get()!= "":
            img = cv2.imread(imgpath_Var[3].get())
        else:
            img = cv2.imread(img_path[3])
    if prediction =="Sad":
        if imgpath_Var[4].get()!= "":
            img = cv2.imread(imgpath_Var[4].get())
        else:
            img = cv2.imread(img_path[4])
    if prediction =="Surprise":
        if imgpath_Var[5].get()!= "":
            img = cv2.imread(imgpath_Var[5].get())
        else:
            img = cv2.imread(img_path[5])
    if prediction =="Neutral":
        if imgpath_Var[6].get()!= "":
            img = cv2.imread(imgpath_Var[6].get())
        else:
            img = cv2.imread(img_path[6])

    img_wh = int(width/2)
    img = cv2.resize(img, (img_wh,img_wh))

    img_top_left = (int(top_left[0]-img_wh),int(top_left[1]))
    img_bottom_right = (int(top_left[0]),int(top_left[1]+img_wh))

    if img_top_left[0]>0 and img_top_left[1]>0:
        img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        _, mask = cv2.threshold(img_gray, 25, 255, cv2.THRESH_BINARY_INV)
```

```
        area = frame[img_top_left[1]: img_top_left[1]+img_wh, img_top_left
[0]: img_top_left[0]+img_wh]
        area_no = cv2.bitwise_and(area, area, mask = mask)
        final = cv2.add(area_no, img)
        frame[img_top_left[1]: img_top_left[1]+img_wh, img_top_left[0]: im
g_top_left[0]+img_wh] = final


    return frame
```