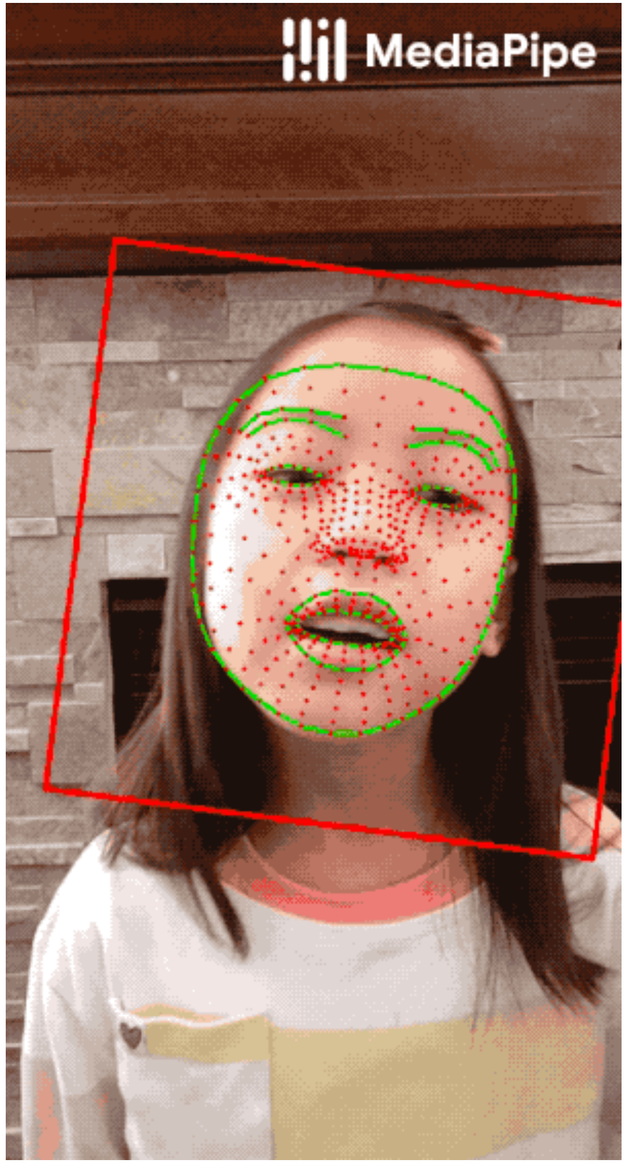
MediaPipe - face mask

211062021 詹凱淇



下載MediaPipe

(終端機

```
$ python3 -m venv mp_env && source mp_env/bin/activate
$ pip install mediapipe
```

https://google.github.io/mediapipe/getting_started/python

Face Mesh - 圖片

匯入套件

```
import cv2
import mediapipe as mp
```

引入mediapipe的功能

```
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_face_mesh = mp.solutions.face_mesh
```

→ 繪圖方法

→ 繪圖樣式

→ 人臉網格標誌

指定圖片位置

```
IMAGE_FILES = [“檔案位置”]
```

→ 可一次放入多張圖片

設定繪圖參數

```
drawing_spec = mp_drawing.DrawingSpec(thickness=1, circle_radius=1)
```

網格粗度　　節點半徑

https://google.github.io/mediapipe/getting_started/python

Face Mesh - 圖片

人臉網格選項設定

```
with mp_face_mesh.FaceMesh(
    static_image_mode=True,
    max_num_faces=5,
    refine_landmarks=True,
    min_detection_confidence=0.5) as face_mesh:
```

- static_image_mode=True → 靜態影像模式
- max_num_faces=5 → 最多可偵測的臉數
- refine_landmarks=True → 細部偵測功能（眼、唇）
- min_detection_confidence=0.5 → 偵測信心度，介於 0-1

取得人臉網格資訊

```
for idx, file in enumerate(IMAGE_FILES):
    image = cv2.imread(file)
    results = face_mesh.process(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
```

- for idx, file in enumerate(IMAGE_FILES): → 根據“圖片數量”迴圈
- cv2.cvtColor(image, cv2.COLOR_BGR2RGB)) → BGR 轉 RGB

判斷是否有結果

```
if not results.multi_face_landmarks:
  continue
```

- if not results.multi_face_landmarks: → 若無臉部資訊，程式結束

複製原圖

```
annotated_image = image.copy()
```

https://google.github.io/mediapipe/getting_started/python

將結果的座標印出來

```
for face_landmarks in results.multi_face_landmarks:
      print('face_landmarks:', face_landmarks)
```

根據“臉部數量”迴圈

將結果標註在影像上

```
      mp_drawing.draw_landmarks(
          image=annotated_image,
          landmark_list=face_landmarks,
          connections=mp_face_mesh.FACEMESH_TESSELATION,
          landmark_drawing_spec=None,
          connection_drawing_spec=mp_drawing_styles
          .get_default_face_mesh_tesselation_style())
```

對象指定為複製影像

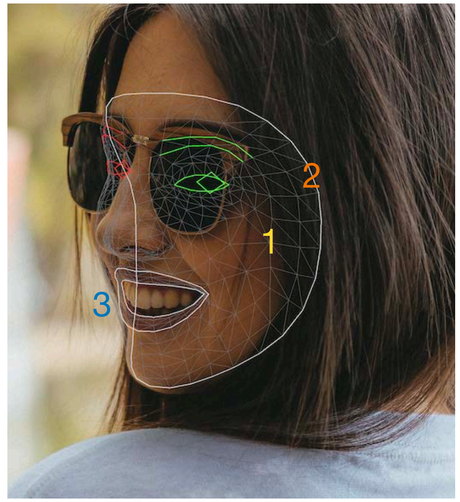
1

```
      mp_drawing.draw_landmarks(
          image=annotated_image,
          landmark_list=face_landmarks,
          connections=mp_face_mesh.FACEMESH_CONTOURS,
          landmark_drawing_spec=None,
          connection_drawing_spec=mp_drawing_styles
          .get_default_face_mesh_contours_style())
```

2

```
      mp_drawing.draw_landmarks(
          image=annotated_image,
          landmark_list=face_landmarks,
          connections=mp_face_mesh.FACEMESH_IRISES,
          landmark_drawing_spec=None,
          connection_drawing_spec=mp_drawing_styles
          .get_default_face_mesh_iris_connections_style())
```

3



Face Mesh - 圖片

輸出圖片

```
cv2.imwrite('我想放這裡' + str(idx) + '.png', annotated_image)
```

設定要儲存的位置

https://google.github.io/mediapipe/getting_started/python

Face Mesh - 相機

```
import cv2
import mediapipe as mp

mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_face_mesh = mp.solutions.face_mesh

IMAGE_FILES = [“檔案位置”]
drawing_spec = mp_drawing.DrawingSpec(thickness=1, circle_radius=1)

cap = cv2.VideoCapture(0)

with mp_face_mesh.FaceMesh(
    static_image_mode=True,
    max_num_faces=5,
    refine_landmarks=True,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as face_mesh:
```

引入mediapipe的功能

連接網路攝影機

設定繪圖參數

追蹤信心度，介於 0-1

Face Mesh - 相機

讀取攝影機畫面

取得人臉網格資訊

影像轉換回BGR

```
while cap.isOpened():
    success, image = cap.read()
    if not success:
      print("Ignoring empty camera frame.")
      continue

    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    results = face_mesh.process(image)

    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
```

檢測影像是否讀取成功

將圖片標記為不可寫入，節省效能

BGR 轉 RGB

RGB 轉 BGR

Face Mesh - 相機

將結果標註在影像上

```
if results.multi_face_landmarks:
    for face_landmarks in results.multi_face_landmarks:
        print('face_landmarks:', face_landmarks)
        mp_drawing.draw_landmarks(
        ...
            .get_default_face_mesh_iris_connections_style()
            )
```

顯示影像、設定關閉條件

```
cv2.imshow('MediaPipe Face Mesh', cv2.flip(image, 1))
```

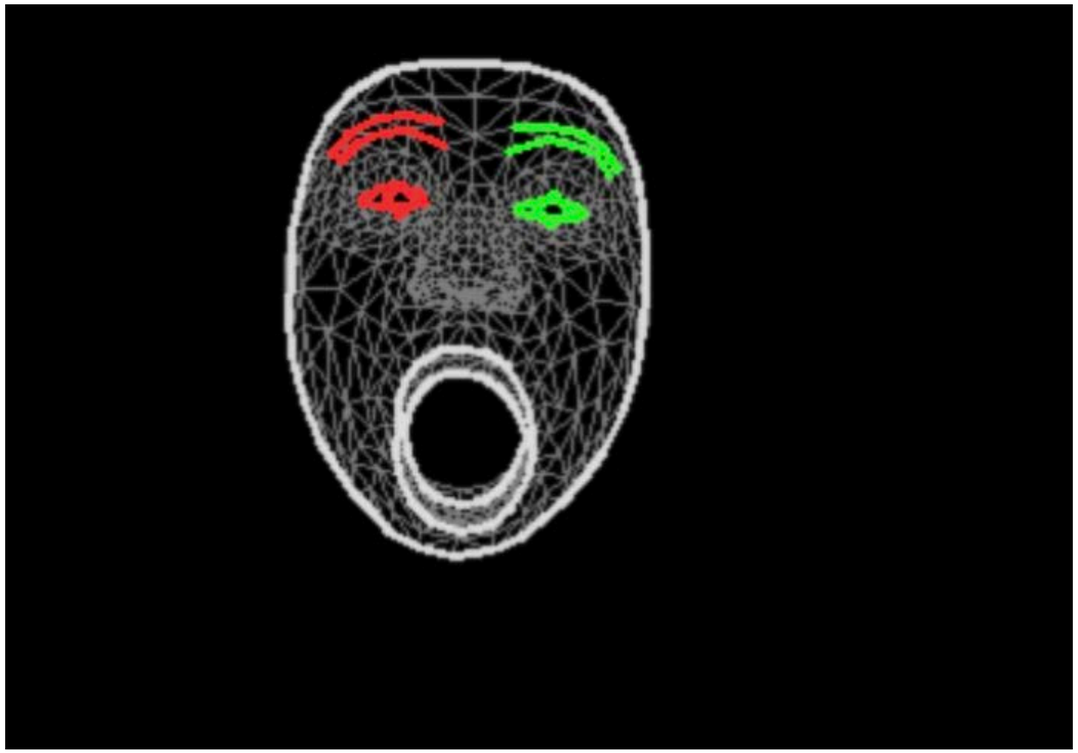
→ 影像左右翻轉

```
if cv2.waitKey(5) & 0xFF == 27:
    break
```

→ 按下 Esc 鍵關閉

```
cap.release()
```

Face Mesh - 相機2（消失的人）



https://steam.oxxostudio.tw/category/python/ai/ai-mediapipe-face-mesh.html

參考資料

https://www.circuspi.com/index.php/2021/06/23/ai-mediapipe-unit3/

https://steam.oxxostudio.tw/category/python/ai/ai-mediapipe-face-mesh.html