

# Homework#2

---

姓名：黃楚祐  
學號：00557043  
日期：2019/6/9

## 方法

吃 pizza，使用宇智波隱形躲過障礙物。

### 1. 先讀背景

```
125 def loadBackground(cap): ##讀取背景
126     loadBackgroundCount = 0
127     ret, frame = cap.read()
128     if ret==False:
129         return frame
130     while loadBackgroundCount < 20: ##需要20次畫面中都沒有動作才能被判斷為背景
131         ret, frame = cap.read()
132         frame = cv2.flip(frame, 1)
133         if ret==False:
134             return frame
135         gbg = fgbg.apply(frame)
136         mvAreaRate = calMovingAreaRate(gbg)
137         if(mvAreaRate > lowerT): ##如果有動作就在數一次
138             loadBackgroundCount = 0
139         else:
140             loadBackgroundCount+=1
141         cv2.imshow('cam', gbg)
142         key = cv2.waitKey(20) & 0xFF
143         if key == 27:
144             return frame
145     return frame
```

### 2. 讀前景畫面後偵測臉和眼的位置並回傳

```
169     facesPos, eyesPos = detectFace(frame) ##偵測臉、眼位置並回傳
```

### 3. detectFace 函式(步驟 3~5): 先轉成灰階後偵測臉的位置

```
62 def detectFace(frame):
63     """ Input = frame from video stream
64         Output = Image with rectangle box in the face
65     """
66
67     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
68
69     # Now get the tuples that detect the faces using above cascade
70     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
71     # faces are the tuples of 4 numbers
72     # x,y => upperLeft corner coordinates of face
73     # width(w) of rectangle in the face
74     # height(h) of rectangle in the face
75     # grey means the input image to the detector
76     # 1.3 is the kernel size or size of image reduced when applying the detection
77     # 5 is the number of neighbors after which we accept that is a face
78
79     i = 0
80
81     face0 = np.zeros(0)
82     face1 = np.zeros(0)
83     facesPos = np.zeros((10,4))
84     eyesPos = np.zeros((10,2,4))
```

#### 4. 記錄每張臉的位置後針對每張臉在偵測眼睛的位置並記錄

```
87 for (x,y,w,h) in faces:
88     cv2.rectangle(frame, (x,y), (x+w, y+h), (255,0,0), 2)
89     # Arguments => image, top-left coordinates, bottomright coordinates, color, rectangle border thickness
90
91     facesPos[i][0] = y
92     facesPos[i][1] = h
93     facesPos[i][2] = x
94     facesPos[i][3] = w ##記錄各張臉的位置
95     if i==0:
96         face0 = frame[y:y+h, x:x+w, :].copy()
97     if i==1:
98         face1 = frame[y:y+h, x:x+w, :].copy() ##紀錄兩張臉的樣子，以便之後換臉
99     # we now need two region of interests(ROI) grey and color for eyes one to detect and another to draw rectangle
100     roi_gray = gray[y:y+h, x:x+w]
101     roi_color = frame[y:y+h, x:x+w]
102     # Detect eyes now
103     eyes = eyes_cascade.detectMultiScale(roi_gray, 1.2, 3)
104     j = 0
105     for (ex, ey, ew, eh) in eyes:
106         if j > 1: #一次只有一雙眼
107             break
108         eyesPos[i][j][0] = ey + y
109         eyesPos[i][j][1] = eh
110         eyesPos[i][j][2] = ex + x
111         eyesPos[i][j][3] = ew ##記錄每個眼睛的位置
112         j+=1
113     i+=1
```

#### 5. 如果出現第二個人就換臉

```
115 facesPos = np.array(facesPos, dtype=np.int)
116 eyesPos = np.array(eyesPos, dtype=np.int) ##必須是int型態才能resize
117 if (face1.ndim == 3): ##當出現第二張臉時，換臉
118     face1 = cv2.resize(face1, (int(facesPos[0][1]), int(facesPos[0][3])))
119     face0 = cv2.resize(face0, (int(facesPos[1][1]), int(facesPos[1][3])))
120     frame[facesPos[0][0]:facesPos[0][0]+facesPos[0][1], facesPos[0][2]:facesPos[0][2]+facesPos[0][3]] = face1
121     frame[facesPos[1][0]:facesPos[1][0]+facesPos[1][1], facesPos[1][2]:facesPos[1][2]+facesPos[1][3]] = face0
122
123     return facesPos, eyesPos
```

#### 6. 先計算前景的移動率，動作比較小就字智波隱形

在背景畫出血輪眼，最後把前景後景 overlap，製造隱形效果

```
172 ##使用字智波隱形
173 if (mvAreaRate < 1): #當人的移動率小於一時，開啟寫輪眼
174     for es in eyesPos:
175         for e in es:
176             if (e[1] != 0 or e[3] != 0):
177                 fakeEyeT = cv2.resize(fakeEye.copy(), (e[1], e[3]))
178                 #cv2.rectangle(backgroundT, (e[2],e[0]), (e[2]+e[3], e[0]+e[1]), (0, 50, 255), -1)
179                 backgroundT[e[0]:e[0]+e[1], e[2]:e[2]+e[3]] = fakeEyeT
180     overlapping = cv2.addWeighted(backgroundT, 1-mvAreaRate, frame, mvAreaRate, 0)
```

#### 7. 計算前景率的方法，計算前景面積後除以高門檻值

```
53 def calMovingAreaRate(img): ##傳入backgroundsubtractor的圖片，計算前景面積
54     mvArea = np.count_nonzero(img == 255)
55     if mvArea > higherT:
56         return 1 #移動面積超過高門檻判定為移動
57     elif mvArea < lowerT:
58         return 0 #移動面積低過低門檻判定為隱形
59     return (mvArea - lowerT) / higherT |
```

8. 使用嘴巴的位置(一個點)，判斷是否碰到 pizza 的範圍，並畫出

pizza

如果吃到 pizza 就縮小 pizza

```
182     eat = droppingPizza(overlapping, pizzaPos, col, pizzaH, facesPos) ##以嘴巴位置判斷是否吃到pizza
183     if(eat):
184         score += speed
185         pizzaH -= speed #被吃的pizza變小
```

9. 放出 bridge 干擾人，如果人臉的範圍碰到 bridge 的範圍就 game

over，此時可用字智波隱形避開

```
188 if(score % (speed * 6 * 3) == 0 and score != 0): ##當分數到達門檻，叫出bridge干擾人
189     isBridgeDropping = True
190 if(bridgePos + bridge.shape[0] == frame.shape[0]):#當bridge的位置到達畫面底部時，需要等到下一次觸發bridge dropping
191     bridgePos = 0
192     isBridgeDropping = False
193 if(isBridgeDropping): #橋下來，並判斷人臉式不是撞到橋
194     hit = droppingBridge(overlapping, bridgePos, facesPos)
195     if(hit and mvAreaRate == 1): ##如果撞到bridge，且沒有用血輪眼進行虛化就 game over
196         cv2.putText(overlapping, "Game Over", (0, frame.shape[1] // 2 - 40), cv2.FONT_HERSHEY_SIMPLEX, 3.65, (0, 0, 255), 10,
197         cv2.putText(overlapping, "Score: %d"%score, (160, frame.shape[1] // 2 + 40), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255)
198         cv2.imshow('cam', overlapping)
199         cv2.waitKey()
200         break ##End
201     bridgePos+=speed // 6
```

10. 最後畫出影像並根據條件改變 pizza 的位置

```
204     cv2.putText(overlapping, str(score), (0, 25), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA) #秀分數
205     cv2.imshow('cam', overlapping)
206     key = cv2.waitKey(20) & 0xFF
207     if key == 27:
208         break
209     pizzaPos+=speed // 3
210     pizzaPos = pizzaPos % (frame.shape[0] - pizza.shape[0]) #pizza到底部時回到最上面
211     if(pizzaH <= 0): #pizza被吃完時回到最上面
212         pizzaPos = 0
```

結果

結果影片：

單人 <https://youtu.be/8jJeyn6BmsU>

雙人 <https://youtu.be/FUWGhSdODPQ>

## 結論

偵測臉、眼的地方用了網路上提供的方法，雖然用起來不太踏實但也很有趣，過程中順便做了換臉的功能。

## 參考文獻

即時臉部偵測

<https://blog.goodaudience.com/real-time-face-and-eyes-detection-with-opencv-54d9ccfee6a8>