1. Code 部分

```
1. /*作業4: 人臉區域處理*/
2.  #include "opencv2/opencv.hpp"
3.  #include <iostream>
4.  #include <time.h>
5.
6.  using namespace cv; //宣告 opencv 函式庫的命名空間
7.  using namespace std; //宣告 C++函式庫的命名空間
8.
9.  /** Function Headers */
10. void detectAndDisplay(void); //偵測人臉的函式宣告
11.
12. /** Global variables */
13. String face_cascade_name =
    "C:/Users/Zoe/source/repos/Project3/data/haarcascade_frontalface_alt.xml";//正面人
    臉瀑布偵測訓練數據
14. //放專案下："haarcascade_frontalface_alt.xml"(跟sln檔放一起)
15. //相對路徑："data/haarcascade_frontalface_alt.xml"(放在下一層的data檔案夾)
16. //絕對路徑："D: / AAA / BBB / CCC / haarcascade_frontalface_alt.xml"(任意位置)
17.
18. CascadeClassifier face_cascade; //建立瀑布分類器物件
19.
20. Mat im, im2, im3;//輸入影像
21. int c;
22. int option=3, width, height;
23. bool ldown = false, lup = false;
24. Point corner;
25.
26.
27. //定義滑鼠反應函式 mouse_callback
28. static void mouse_callback(int event, int x, int y, int, void *)
29. {
30.   // 當滑鼠按下左鍵，根據點選位置，得到選項 (option) 數值
31.   if (event == EVENT_LBUTTONDOWN);
32.   {
33.         ldown = true;
34.         corner.x = x;
```

```
35.         corner.y = y;
36.         cout << "corner recorde at" << corner << endl;
37.   }
38. }
39.
40. /**
41.  * @function main
42.  */
43. int main( void )
44. {
45.   VideoCapture cap("C:/Users/Zoe/source/repos/Project3/data/sleepy.mpg"); //讀取影片或相機
46.   //VideoCapture cap(0);
47.
48.   //放專案下："sleepy.mpg"(跟sln檔放一起)
49.   //相對路徑："data/sleepy.mpg"(放在下一層的data檔案夾)
50.   //絕對路徑："D: / AAA / BBB / CCC / sleepy.mpg"(任意位置)
51.
52.   if (!cap.isOpened()) return 0; //不能讀視訊的處理
53.
54.      ///匯入人臉瀑布偵測訓練數據
55.   if (!face_cascade.load(face_cascade_name)) { printf("--(!)Error loading face cascade\n"); waitKey(0); return -1; };
56.
57.   while (char(waitKey(1)) != 27 && cap.isOpened()) //當鍵盤沒按 Esc，以及視訊物件成功開啟時，持續執行 while 迴圈
58.      {
59.         cap >> im; //抓取視訊的畫面
60.           if( im.empty() ) //如果沒抓到 畫面
61.           {
62.               printf(" --(!) No captured im -- Break!");  //顯示錯誤訊息
63.               break;
64.           }
65.
66.         //前處理
67.         //讀取視訊的寬 width
68.         cout << "width" << im.cols << endl;
69.         //讀取視訊的高 height
```

```
70.         cout << "height" << im.rows << endl;
71.
72.
73.         //imshow("window", im);
74.         //定義視窗名稱 namedWindow
75.         //namedWindow("window");
76.
77.         //設定滑鼠反應函式 setMouseCallback
78.         setMouseCallback("window", mouse_callback);
79.
80.         /*偵測人臉，並顯示AR圖像融合結果*/
81.           detectAndDisplay();
82.     }
83.     return 0;
84. }
85.
86. /** @function detectAndDisplay */
87. void detectAndDisplay(void)
88. {
89.  //人臉偵測部分
90.  vector<Rect> faces; //建立人臉ROI 向量
91.  Mat im_gray; //灰階影像物件
92.
93.  cvtColor(im, im_gray, COLOR_BGR2GRAY); //彩色影像轉灰階
94.  equalizeHist(im_gray, im_gray); //灰階值方圖等化(對比自動增強)。若視訊品質好，可不
    用
95.
96.  //人臉瀑布偵測
97.  face_cascade.detectMultiScale(im_gray, faces, 1.1, 4, 0, Size(80, 80));
98.
99.  //獲得最大人臉的 ROI數據
100.      if (faces.size() > 0) {
101.          int largest_area = -999;
102.          int largest_i;
103.          for (int i = 0; i < faces.size(); i++) //用迴圈讀取所有人臉 ROI
104.          {
105.              //定義影像中的 ROI
106.              if (largest_area < faces[i].height)
```

```
107.                {
108.                     largest_area = faces[i].height;
109.                     largest_i = i;
110.                }
111.           }
112.
113.           Rect faceROI = faces[largest_i]; //將最大人臉的 ROI數據存入 faceROI
114.
115.           // 稍為放大ROI，使新臉能夠完整覆蓋視訊中的人臉(可不做)
116.           int d = 25;
117.           faceROI.x = faceROI.x - d;
118.           faceROI.y = faceROI.y - d;
119.           faceROI.width = faceROI.width + 2 * d;
120.           faceROI.height = faceROI.height + 2 * d;
121.
122.
123.           //繪製人臉區域矩形框
124.           rectangle(im, Point(faceROI.x, faceROI.y), Point(faceROI.x +
    faceROI.width, faceROI.y + faceROI.height), Scalar(0, 0, 255), 2);;
125.           putText(im, string("M10719005"), Point(faceROI.x, faceROI.y - 10), 0, 1,
    Scalar(0, 255, 255), 3);
126.           im2 = im.clone();
127.           Mat imRoi = im2(Rect(faceROI.x, faceROI.y, faceROI.width ,
    faceROI.height));
128.           //繪製window下方的選項文字
129.           putText(im2, string("Blur"), Point(50, 400), 0, 1, Scalar(0, 255, 255),
    3);
130.           putText(im2, string("Negative"), Point(200, 400), 0, 1, Scalar(255, 0,
    255), 3);
131.           putText(im2, string("Clear"), Point(450, 400), 0, 1, Scalar(128, 128,
    128), 3);
132.           //根據 option 選項，處理ROI影像
133.
134.           if ((corner.y >= 380) && (corner.y <= 450))
135.           {
136.                if ((corner.x >= 50) && (corner.x <= 100))
137.                {
138.
```

```
139.                     blur(imRoi, im3, Size(10,10));
140.                     addWeighted(imRoi, 0, im3, 1, 0, imRoi);
141.                }
142.                else if ((corner.x >= 200) && (corner.x <= 350))
143.                {
144.                     bitwise_not(imRoi, im3);
145.                     addWeighted(imRoi, 0, im3, 1, 0, imRoi);
146.                }
147.                else if ((corner.x >= 450) && (corner.x <= 550))
148.                {
149.                     addWeighted(imRoi, 0, imRoi, 1, 0, imRoi);
150.                }
151.           }
152.
153.           //顯示影像
154.           imshow("window", im2);
155.      }
156. }
157.
```