

### 1. 引入函式庫：

- 引入 OpenCV 函式庫 (`opencv2/opencv.hpp`)。
- 引入 C++ 輸入輸出流函式庫 (`iostream`)。

### 2. 使用命名空間：

- 使用 OpenCV 命名空間 (`cv`)。
- 使用標準命名空間 (`std`)。

### 3. 自訂函式宣告：

- 宣告 `detectAndDisplay` 函式，用於偵測人臉和顯示影像。

### 4. 全域變數宣告：

- `face_cascade_name`：正面人臉哈爾階層式分類器的檔案名稱。
- `eyes_cascade_name`：人眼哈爾階層式分類器的檔案名稱。
- `face_cascade` 和 `eyes_cascade`：建立正面人臉與人眼哈爾階層式分類器物件。
- `im`：輸入影像。
- `option`：特效選項。
- `cap1`：人臉視訊物件。
- `cap2`：特效影片物件。
- `eye_centers`：雙眼中心的位置。
- `click_position`：滑鼠點擊位置。
- `play_explosion`：播放 `explosion.mp4` 的標誌。
- `explosion_frame_count`：`explosion.mp4` 的幀計數器。
- `face_detected`：人臉偵測標誌。

### 5. 滑鼠回調函式：

- `mouse_callback` 函式定義：處理滑鼠事件。
  - 當滑鼠按下左鍵時，根據點擊位置設定 `option`。

- 如果點擊位置在影像下方選項區域，設定特效選項（綠色、瘦臉、馬賽克）。
- 如果偵測到人臉，點擊其他區域顯示眼睛處的圓圈並播放 `explosion.mp4`。

## 6. 主程式：

- 嘗試開啟人臉視訊和 `explosion.mp4`。
- 載入人臉和人眼哈爾階層式分類器。
- 建立顯示影像的視窗並設定滑鼠回調函式。
- 進行主迴圈：
  - 從人臉視訊抓取畫面。
  - 執行 `detectAndDisplay` 函式。
  - 如果需要播放 `explosion.mp4`，抓取特效影片的畫面，並顯示在點擊位置。
  - 顯示影像。

## 7. `detectAndDisplay` 函式：

- 進行人臉偵測前的預處理：
  - 彩色影像轉灰階並進行直方圖均衡化。
  - 使用哈爾階層式分類器檢測正面人臉。
- 如果偵測到人臉：
  - 找出最大的人臉 ROI 區域並稍微擴大。
  - 偵測人眼位置，計算雙眼中心位置。
  - 根據選項處理 ROI 影像：
    - 選項 1：將人臉變綠色。
    - 選項 2：實現瘦臉效果。
    - 選項 3：對人臉進行馬賽克處理。
    - 選項 4：畫眼睛到滑鼠點擊位置的直線。
  - 繪製人臉區域矩形框以及學號。
- 顯示影像。