嵌入式系統設計

期末報告

人臉識別門禁系統

第九組
113522107 蕭育傑
113522082 謝昀彤

一、介紹

1. 前言

我們的專案基於 ESP32-CAM 開發板，設計並實現了一套低成本的智慧門禁系統，結合人臉識別技術，提供自動化的門禁控制與即時交互功能。系統採用 HTTP 協定提供靜態網頁，並通過 WebSocket 協定實現實時命令處理與影像傳輸。使用者可在瀏覽器上完成**註冊人臉**、**辨識訪客**、**開門控制**等操作。

本專案的相機模組負責捕捉影像並提取人臉特徵，通過在 local 端比對確保準確性與安全性。若辨識成功，系統會開啟門鎖並回饋信息；若失敗，則回傳錯誤提示。我們的系統以**低成本**、**高效率**為特色，適合智慧家居與中小型企業的門禁需求。

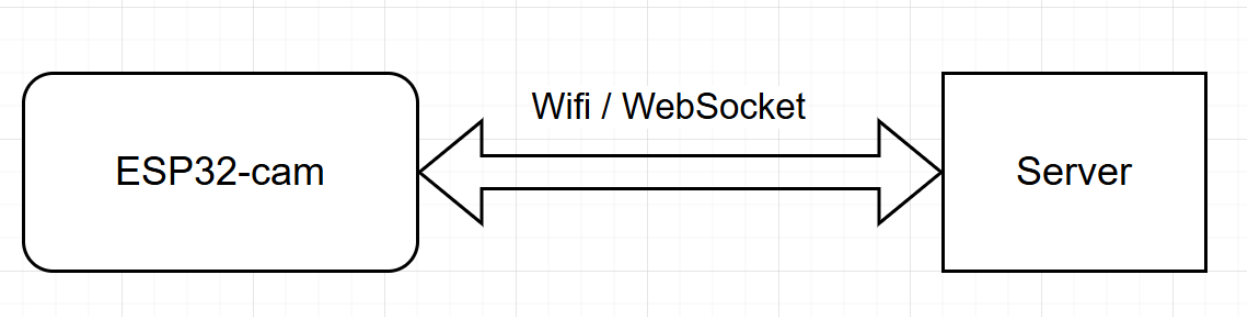
未來這個門禁系統可整合人工智慧、雲端數據管理及多傳感器模組，進一步提升系統的智慧化與可用性，為智慧城市及物聯網領域提供更多創新應用。

2. 系統運作概述

我們的系統在初始階段會先做 Wi-Fi 的連線，接著對相機初始化，再載入之前的人臉資料，最後會建立一個 http server 讓方便使用者管理。使用者進入那個網頁後就可以選取想要執行的動作，如人臉註冊，接著系統就會執行對應的動作，執行完畢後，系統會輸出一些結果讓使用者知道，像是告知使用者這張臉是否有註冊。完成本次的命令要求後系統就會釋放本次所用的資源並進入等待下一個命令的狀態。在之後的章節我們會對一些核心的功能做更詳細的描述。

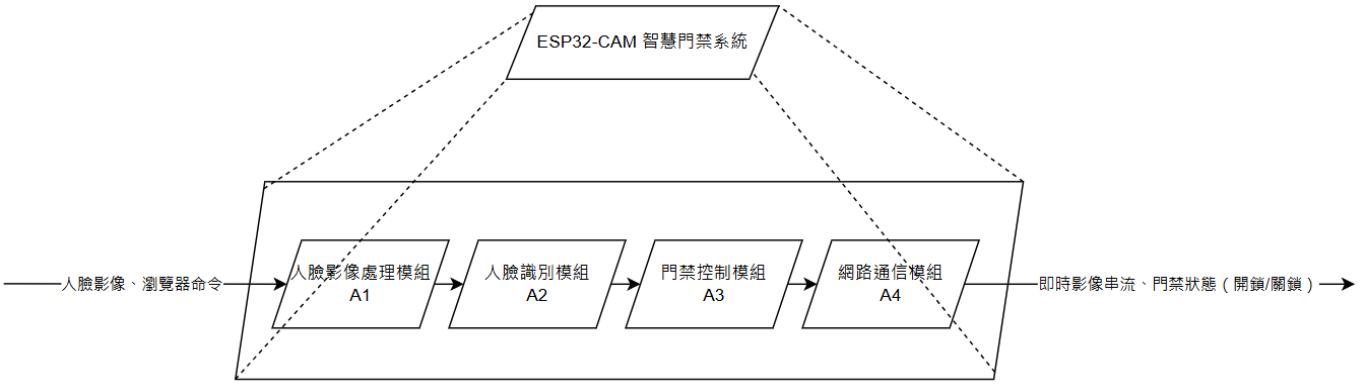
二、系統架構

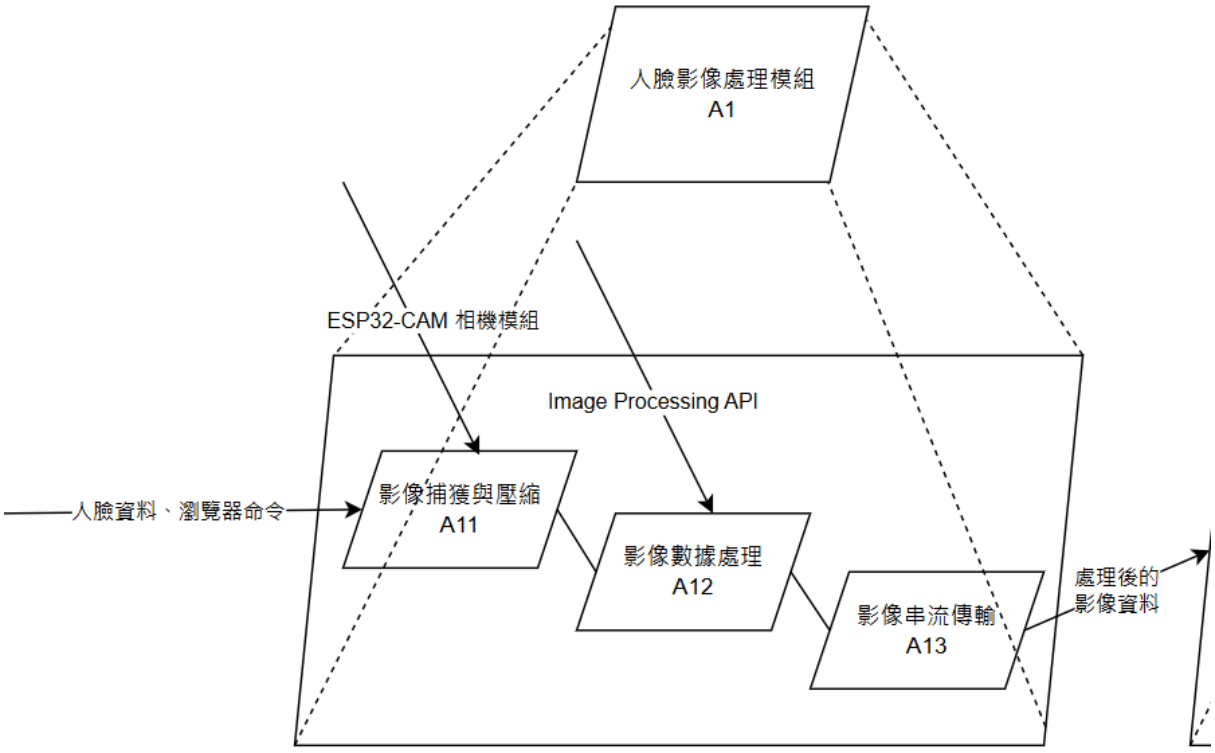
1. 系統架構

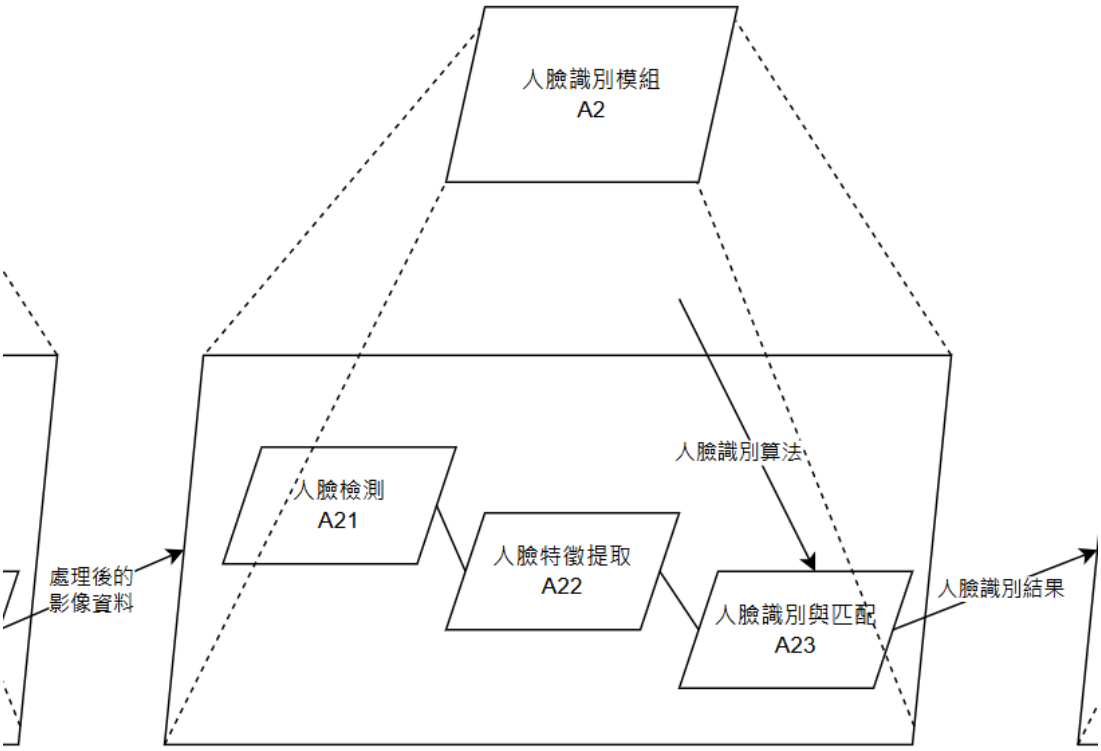


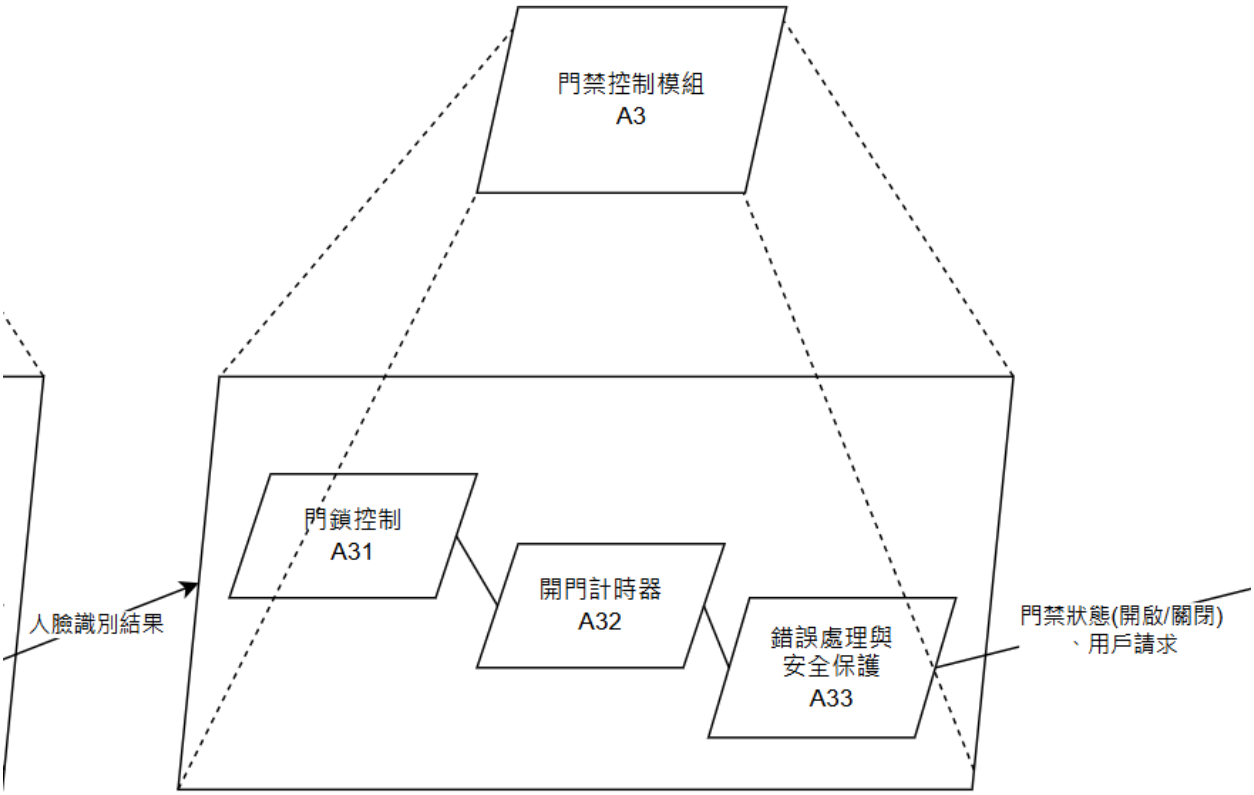
三、系統設計

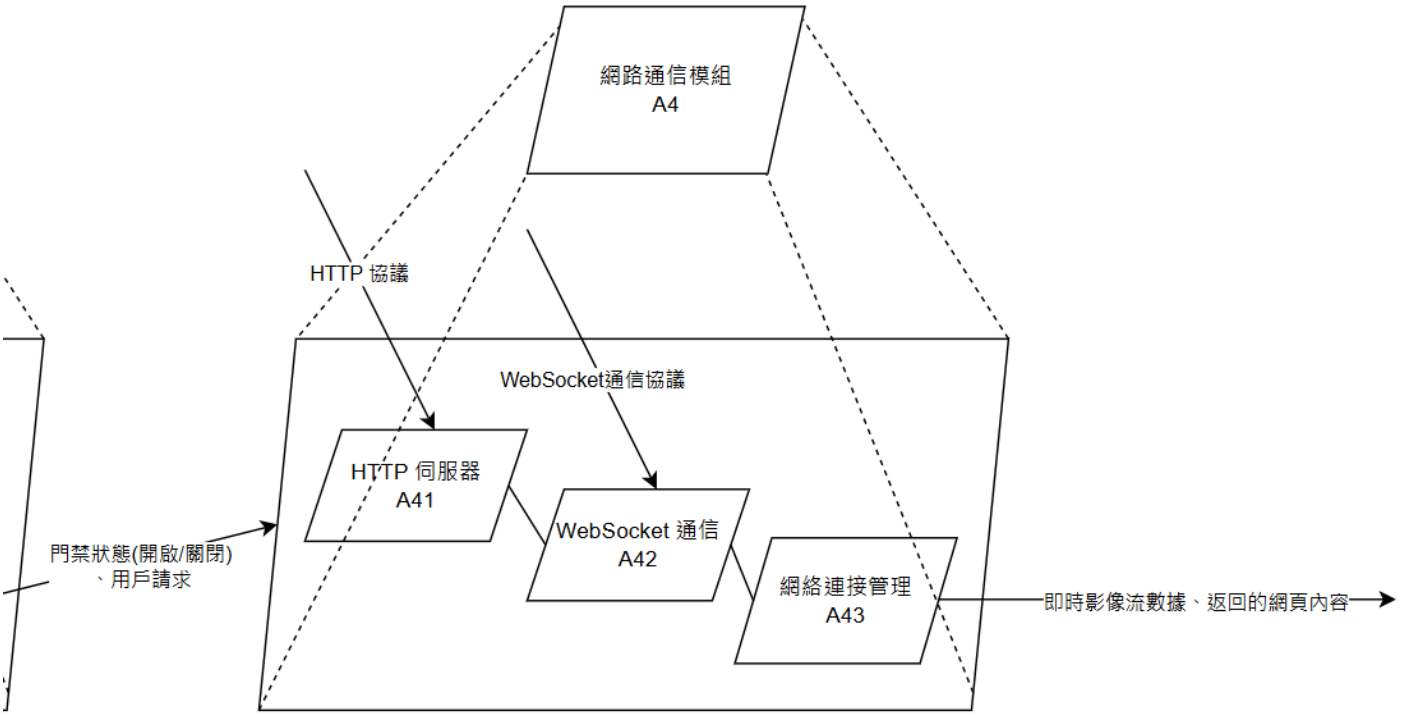
1. IDEF0



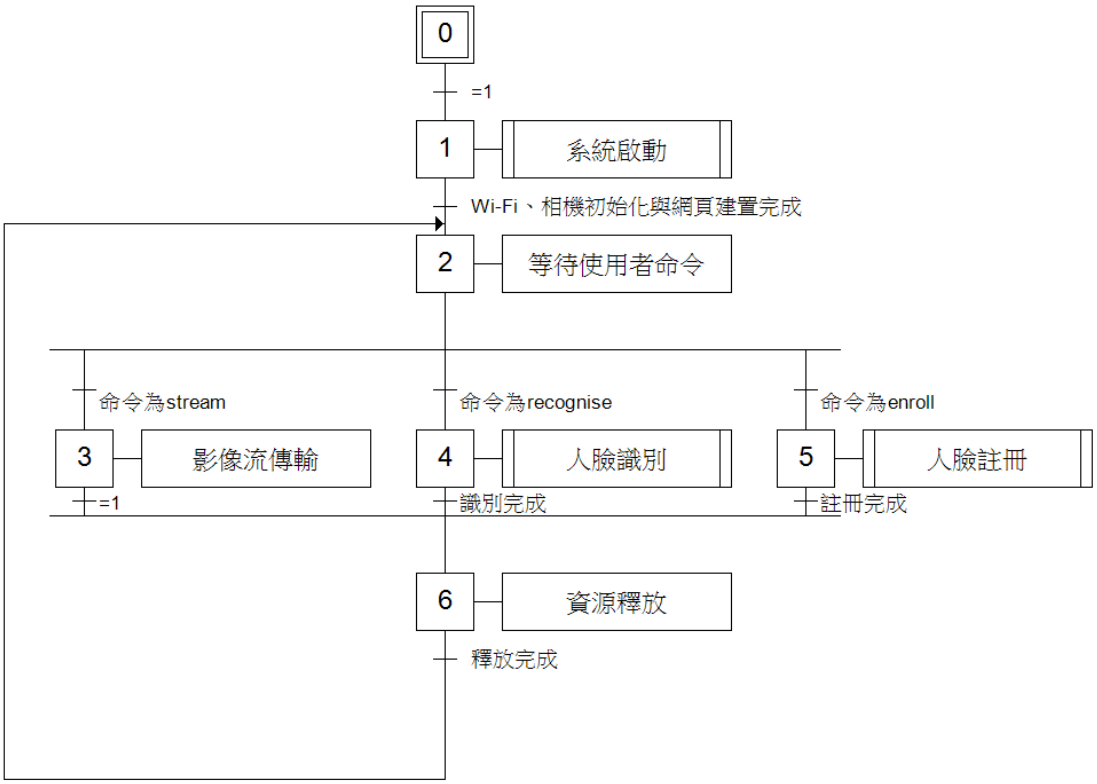


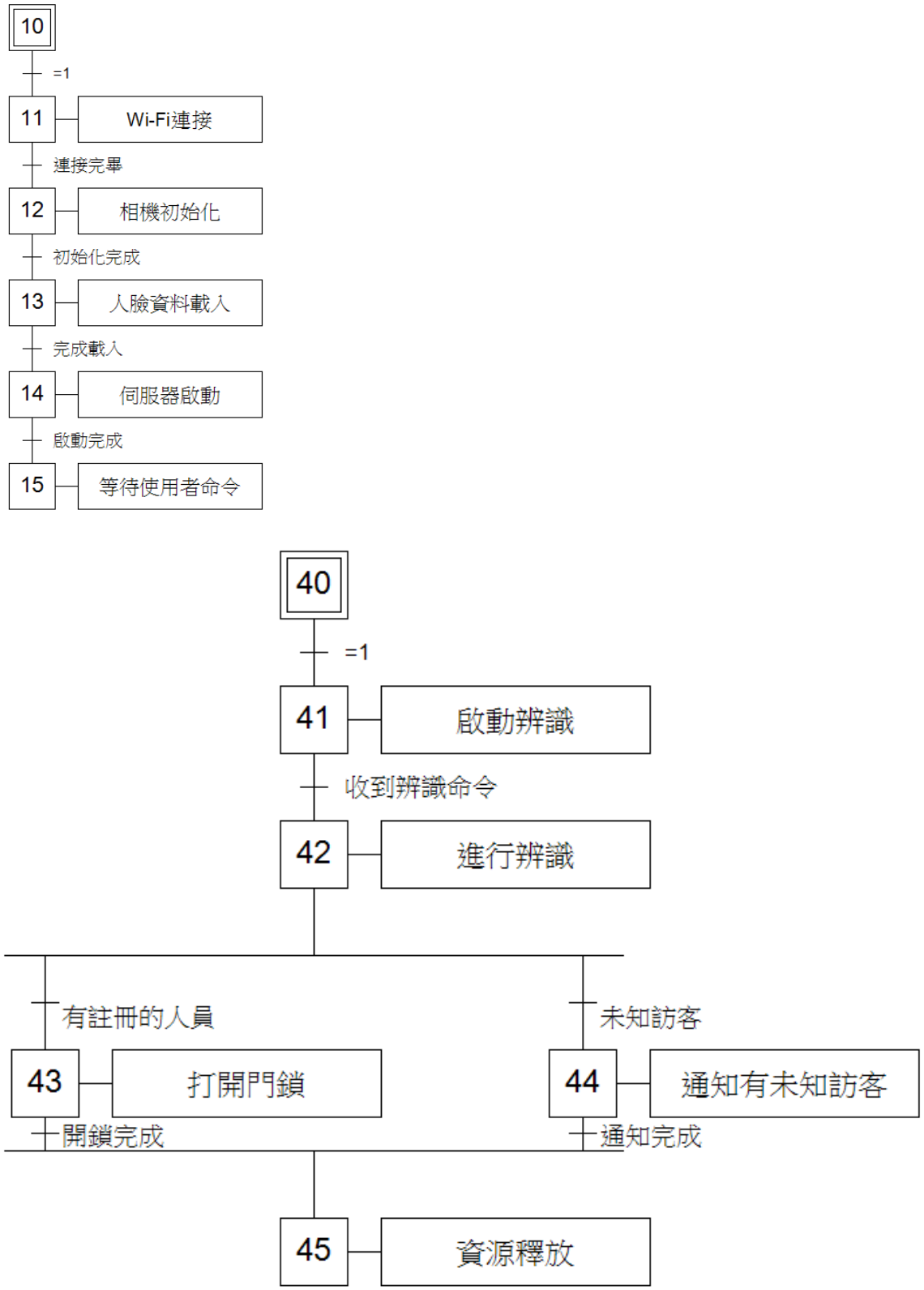

人臉識別模組
A2
人臉識別算法
人臉檢測
A21
人臉特徵提取
A22
人臉識別與匹配
A23
處理後的
影像資料
人臉識別結果

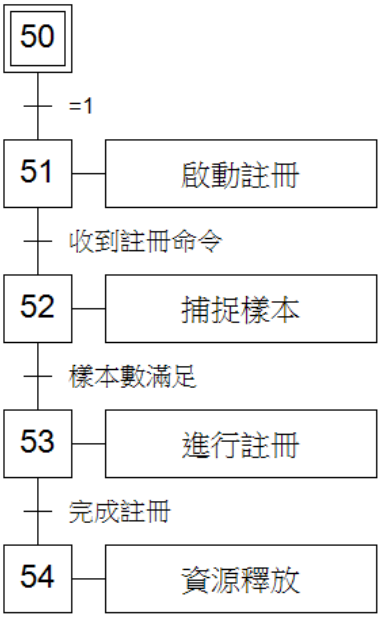

門禁控制模組
A3
門鎖控制
A31
開門計時器
A32
錯誤處理與
安全保護
A33
人臉識別結果
門禁狀態(開啟/關閉)
、用戶請求


網路通信模組
A4
HTTP 協議
WebSocket通信協議
HTTP 伺服器
A41
WebSocket 通信
A42
網絡連接管理
A43
門禁狀態(開啟/關閉)
、用戶請求
即時影像流數據、返回的網頁內容

2. Grafcet


0
=1
1
系統啟動
Wi-Fi、相機初始化與網頁建置完成
2
等待使用者命令
命令為stream
命令為recognise
命令為enroll
3
影像流傳輸
4
人臉識別
5
人臉註冊
=1
識別完成
註冊完成
6
資源釋放
釋放完成

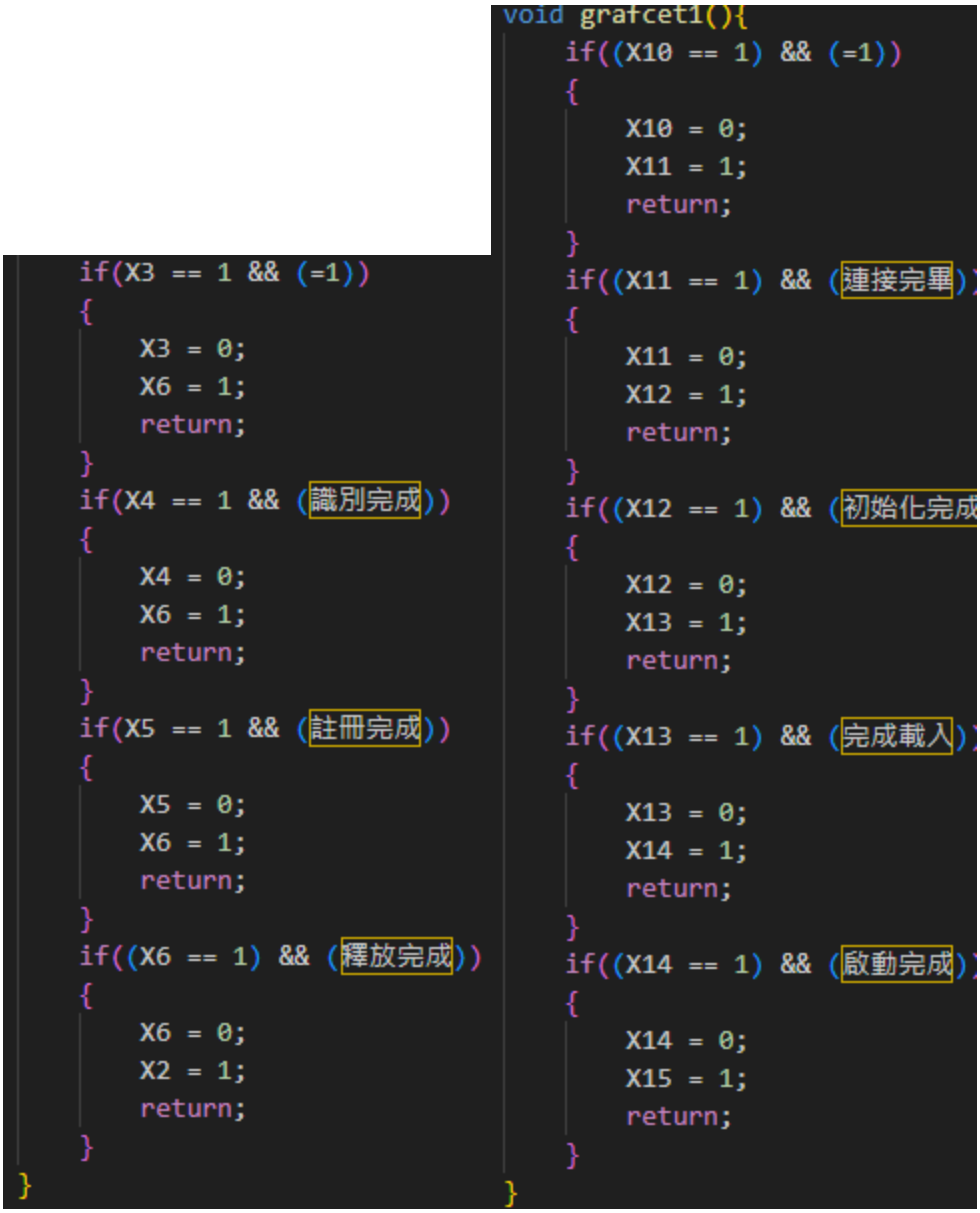

10
=1
11
Wi-Fi連接
連接完畢
12
相機初始化
初始化完成
13
人臉資料載入
完成載入
14
伺服器啟動
啟動完成
15
等待使用者命令
40
=1
41
啟動辨識
收到辨識命令
42
進行辨識
有註冊的人員
未知訪客
43
打開門鎖
44
通知有未知訪客
開鎖完成
通知完成
45
資源釋放



四、高階語言合成

```
#include<stdio.h>
int X0 = 1, X1 = 0, X10 = 1, X11 = 0, X12 = 0, X13 = 0, X14 = 0, X15 = 0, X2 = 0, X3 = 0, X4 = 0, X40 = 1,;
int X41 = 0, X42 = 0, X43 = 0, X44 = 0, X45 = 0, X5 = 0, X50 = 1, X51 = 0, X52 = 0, X53 = 0, X54 = 0, X6 = 0;
void grafcet0();
void datapath0();
void action();
void 系統啟動();
void grafcet1();
void datapath1();
void 等待使用者命令();
void 影像流傳輸();
void 人臉識別();
void grafcet4();
void datapath4();
void 人臉註冊();
void grafcet5();
void datapath5();
void 資源釋放();
void action();
void Wi-Fi連接();
void 相機初始化();
void 人臉資料載入();
void 伺服器啟動();
void 等待使用者命令();
void action();
void 啟動辨識();
void 進行辨識();
void 打開門鎖();
void 通知有未知訪客();
void 資源釋放();
void 啟動註冊();
void 捕捉樣本();
void 進行註冊();
void 資源釋放();
```

```
void main(){
printf("X0 = %d,X1 = %d,X2 = %d,X3 = %d,X4 = %d,X5 = %d,X6 = %d\n",X0 ,X1 ,X2 ,X3 ,X4 ,X5 ,X6 );
while(1){
datapath0();
grafcet0();
printf("X0 = %d,X1 = %d,X2 = %d,X3 = %d,X4 = %d,X5 = %d,X6 = %d\n",X0 ,X1 ,X2 ,X3 ,X4 ,X5 ,X6 );
}
}
void grafcet0(){
if((X0 == 1) && (=1))
    {
        X0 = 0;
        X1 = 1;
        return;
    }
    if((X1 == 1) && (Wi-Fi、相機初始化與網頁建置完成))
    {
        X1 = 0;
        X2 = 1;
        return;
    }
    if(X2 == 1)
    {
        if(命令為stream)
        {
            X2 = 0;
            X3 = 1;
        }
        else if(命令為recognise)
        {
            X2 = 0;
            X4 = 1;
        }
        else if(命令為enroll)
        {
            X2 = 0;
            X5 = 1;
        }
        return;
    }
```



```
    if(X3 == 1 && (=1))
    {
        X3 = 0;
        X6 = 1;
        return;
    }
    if(X4 == 1 && (識別完成))
    {
        X4 = 0;
        X6 = 1;
        return;
    }
    if(X5 == 1 && (註冊完成))
    {
        X5 = 0;
        X6 = 1;
        return;
    }
    if((X6 == 1) && (釋放完成))
    {
        X6 = 0;
        X2 = 1;
        return;
    }
}
```

```
void grafcet1(){
    if((X10 == 1) && (=1))
    {
        X10 = 0;
        X11 = 1;
        return;
    }
    if((X11 == 1) && (連接完畢))
    {
        X11 = 0;
        X12 = 1;
        return;
    }
    if((X12 == 1) && (初始化完成
    {
        X12 = 0;
        X13 = 1;
        return;
    }
    if((X13 == 1) && (完成載入))
    {
        X13 = 0;
        X14 = 1;
        return;
    }
    if((X14 == 1) && (啟動完成))
    {
        X14 = 0;
        X15 = 1;
        return;
    }
}
```

```
void datapath0()
{
if(X0 == 1)
action();
if(X1 == 1)
系統啟動();
if(X2 == 1)
等待使用者命令();
if(X3 == 1)
影像流傳輸();
if(X4 == 1)
人臉識別();
if(X5 == 1)
人臉註冊();
if(X6 == 1)
資源釋放();
}
```

```
void datapath1()
{
if(X10 == 1)
action();
if(X11 == 1)
Wi-Fi連接();
if(X12 == 1)
相機初始化();
if(X13 == 1)
人臉資料載入();
if(X14 == 1)
伺服器啟動();
if(X15 == 1)
等待使用者命令();
}
```

```
void datapath4()
{
if(X40 == 1)
action();
if(X41 == 1)
啟動辨識();
if(X42 == 1)
進行辨識();
if(X43 == 1)
打開門鎖();
if(X44 == 1)
通知有未知訪客();
if(X45 == 1)
資源釋放();
}
```

```
void datapath5()
{
if(X50 == 1);
if(X51 == 1)
啟動註冊();
if(X52 == 1)
捕捉樣本();
if(X53 == 1)
進行註冊();
if(X54 == 1)
資源釋放();
}
```

```
void action(){
printf("action activate !!\n");
}
void 系統啟動()
{
printf("系統啟動 activate !!\n");
datapath1();
grafcet1();
printf("X10 = %d,X11 = %d,X12 = %d,X13 = %d,X14 = %d,X15 = %d\n",X10 ,X11 ,X12 ,X13 ,X14 ,X15 );
}
void 等待使用者命令()
{
printf("等待使用者命令 activate !!\n");
}
void 影像流傳輸()
{
printf("影像流傳輸 activate !!\n");
}
void 人臉識別()
{
printf("人臉識別 activate !!\n");
datapath4();
grafcet4();
printf("X40 = %d,X41 = %d,X42 = %d,X43 = %d,X44 = %d,X45 = %d\n",X40 ,X41 ,X42 ,X43 ,X44 ,X45 );
}
void 人臉註冊()
{
printf("人臉註冊 activate !!\n");
datapath5();
grafcet5();
printf("X50 = %d,X51 = %d,X52 = %d,X53 = %d,X54 = %d\n",X50 ,X51 ,X52 ,X53 ,X54 );
}
```

```
void 資源釋放()
{
printf("資源釋放 activate !!\n");
}
void action()
{
printf("action activate !!\n");
}
void Wi-Fi連接()
{
printf("Wi-Fi連接 activate !!\n");
}
void 相機初始化()
{
printf("相機初始化 activate !!\n");
}
void 人臉資料載入()
{
printf("人臉資料載入 activate !!\n");
}
void 伺服器啟動()
{
printf("伺服器啟動 activate !!\n");
}
void 等待使用者命令()
{
printf("等待使用者命令 activate !!\n");
}
```

```
void action()
{
printf("action activate !!\n");
}
void Wi-Fi連接()
{
printf("Wi-Fi連接 activate !!\n");
}
void 相機初始化()
{
printf("相機初始化 activate !!\n");
}
void 人臉資料載入()
{
printf("人臉資料載入 activate !!\n");
}
void 伺服器啟動()
{
printf("伺服器啟動 activate !!\n");
}
void 等待使用者命令()
{
printf("等待使用者命令 activate !!\n");
}
void action()
{
printf("action activate !!\n");
}
void 啟動辨識()
{
printf("啟動辨識 activate !!\n");
}
void 進行辨識()
{
printf("進行辨識 activate !!\n");
}
```

```
void 打開門鎖()
{
printf("打開門鎖 activate !!\n");
}
void 通知有未知訪客()
{
printf("通知有未知訪客 activate !!\n");
}
void 資源釋放()
{
printf("資源釋放 activate !!\n");
}
void ()
{
printf(" activate !!\n");
}
```

```
void 啟動註冊()
{
printf("啟動註冊 activate !!\n");
}
void 捕捉樣本()
{
printf("捕捉樣本 activate !!\n");
}
void 進行註冊()
{
printf("進行註冊 activate !!\n");
}
void 資源釋放()
{
printf("資源釋放 activate !!\n");
}
```

五、重要程式碼

- Wifi 初始化

```
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
```

上面的程式碼會負責讓 esp32-cam 連線

- 鏡頭初始化

```
// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}
```

這邊會負責做鏡頭的初始化

- 啟動伺服器

```
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_QVGA);
app_httpserver_init();
app_facenet_main();
socket_server.listen(82);

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
```

這邊負責伺服器的建立

- 人臉辨識

```
if (g_state == START_RECOGNITION  && (st_face_list.count > 0))
{
  face_id_node *f = recognize_face_with_name(&st_face_list, out_res.face_id);
  if (f)
  {
    char recognised_message[64];
    sprintf(recognised_message, "DOOR OPEN FOR %s", f->id_name);
    open_door(client);
    client.send(recognised_message);
  }
  else
  {
    client.send("FACE NOT RECOGNISED");
  }
}
```

這邊的 code 是進行人臉辨識的主要 code，其中的

st_face_list: 用來儲存所註冊的人臉資料

out_res.face_id: 是用來儲存人臉特徵

因此透過呼叫 **recognize_face_with_name()**，可以讓目前的人臉與有註冊的人臉做比對，若是有註冊的人臉就輸出是哪個人把門打開的，並呼叫 open_door()把門打開；若是沒有註冊的人臉就會輸出這個人員沒有註冊。

- 人臉註冊

```
if (g_state == START_ENROLL)
{
  int left_sample_face = do_enrollment(&st_face_list, out_res.face_id);
  char enrolling_message[64];
  sprintf(enrolling_message,
  "SAMPLE NUMBER %d FOR %s", ENROLL_CONFIRM_TIMES - left_sample_face, st_name.enroll_name);
  client.send(enrolling_message);
  if (left_sample_face == 0)
  {
    ESP_LOGI(TAG, "Enrolled Face ID: %s", st_face_list.tail->id_name);
    g_state = START_STREAM;
    char captured_message[64];
    sprintf(captured_message, "FACE CAPTURED FOR %s", st_face_list.tail->id_name);
    client.send(captured_message);
    send_face_list(client);

  }
}
```

上面這段 code 是進行註冊的主要 code，其中的

st_face_list:用來儲存所註冊的人臉資料

left_sample_face:用來表示還需要的樣本數

在取得足夠數量的樣本數後，系統就會幫這個人臉註冊，並把這張人臉的代號設為使用者所輸入的代號，方便管理者辨別。

六、未來展望

1. 加入 AI 模型

我們目前的人臉辨識並沒有加入一些 ML 的算法，只是單純的比對一些生物特徵，未來可以嘗試加入 TinyML，讓這個門禁系統在辨識人臉更加的精確。

2. 雲端資料庫

目前我們會把所註冊的人臉儲存在 esp32 cam 的 flash 中，會導致實際可以註冊的人數可能連一個家庭的人數都會有點吃緊，因此若能把資料改放在雲端上，就可以突破人員數的限制，甚至要把這個系統架在一個大樓也是可行的。

3. 擴展應用場景

如上一點所提到，我們目前的系統主要是提供給小家庭做使用，在改善軟硬體後，我們的系統可以在公司或是大型的公寓做使用。

4. 增強資料安全與隱私保護

目前的資料的傳輸是沒有加密的，在充斥著網路攻擊的現代社會來說是非常的不安全，特別是儲存的是人臉如此重要的資料，因此在傳輸上進行加密算法是可以改進的點。如果資料是放在雲端上，也必須對資料加密，以防止未經授權的訪問。