

中華民國第 62 屆中小學科學展覽會

作品說明書

國中組 生活與應用科學(一)科

032814

紅外線體溫量測智慧登錄系統之開發與研究

學校名稱：高雄市立文府國民中學

作者： 國二 何宥澤 國二 邱楷倫 國二 廖宥晴	指導老師： 郭懿尹
---	------------------

關鍵詞：紅外線體溫量測、影像辨識、Python

摘要

自從 COVID-19 疫情爆發以來，學校必須每天監控學生的體溫與身體狀況，花費大量額外的人力與時間。本研究使用 Python 執行 OpenCV 程式，來抓取即時影像中學生的學號資訊，執行 easyocr 判斷學號資訊並比對學生身份，當學生身份確認後，使用 Arduino 與 GY-MCU90615 量測學生體溫，將量測到的體溫回傳至 Python 程式，透過程式將數據進行整合，最後將學生的個人資訊(班級、姓名、座號等)與體溫數值進行登錄後傳至網路平台，產生出體溫監控表單，稱此裝置為『紅外線體溫量測智慧登錄系統』。本研究成功使用(去背+Canny)偵測+彩色辨識的方式，使紅外線體溫量測智慧登錄系統有更佳的辨識成功率與更快的辨識時間，可以快速辨識學生身份，並登錄其體溫，可減少學校防疫的工作量，達到省時省人力的目的。

壹、前言

一、研究動機

自從 COVID-19 疫情爆發以來，人類的生活形態被迫改變，勤洗手、戴口罩已是全世界人們的生活日常。在台灣，因為嚴格的防疫政策，除了出門要戴口罩外，進出商店、百貨公司和餐廳等公共場所，都必須先量測體溫跟實聯制，才可進入。而學校行政單位也必須每天監控學生的體溫與身體狀況，並定時完成學校環境的消毒作業，這些政策雖然有很好的防疫成效，但同時也需要花費大量額外的人力與時間。因此我們開始思考是否可以設計出一個可以體溫量測裝置，來解決每天學生體溫監控過程中，所消耗大量時間與人力的問題。透過相關研究開發出這樣的裝置，並將其運用於校園中，就是我們做這個研究的最大動機。

二、研究背景

在校園的防疫工作中，與我們學生最直接相關的就是體溫的量測與體溫的記錄，目前學校內學生體溫的監控狀況為：每日上午與中午，由師長或是班級幹部進行全班體溫的量測與記錄，完成後將所量測的結果送至學校學務處，學務處行政人員進行學生體溫的登錄與監控，每天必須完成兩次(早上一次，中午一次)體溫監控流程。而這個體溫監控流程有以下缺點：

1. 耗費人力：每次必須有專人幫忙量體溫，若師長或是幹部忙碌，量測時間延後，會影響上課時間或是午休時間。
2. 量測費時：體溫量測完成後，需停下來書寫溫度記錄，在進行下一位學生的體溫量測，平均一位同學的量測時間約為20秒至30秒，整班量測約需花15分鐘至20分鐘。
3. 體溫登錄易誤植：當我們把量測完成的體溫記錄送至學務處，學務處行政人員將把體溫記錄登錄於電腦表單中，但人數眾多，常易誤植量測結果，因此登錄完成須再次確認。

根據上述的體溫監控流程缺點，我們開始思考如何設計出一個可以自動辨識學生身份並進行體溫量測與登錄的裝置，來解決每天學生體溫監控過程中所遇到的問題。

三、研究目的

為了解決上述體溫監控流程的缺點，並設計出一套可以自動辨識學生身份與體溫量測登錄的裝置，我們將此設計概念分成兩大部分進行討論，分別為 1.學生身份辨識 2.學生體溫量測與登錄。

首先，我們針對學生身份辨識的部分與老師討論，討論結果發現，學生在校園中都必須穿著校服，而校服都必須繡上學號，且每位學生的學號都不同，一個學號就代表一個學生的身份，所以只要能夠辨識學號，就能辨識學生的身份。

接著我們也針對學生體溫量測方式與老師進行討論，而老師也提醒我們，因為 COVID-19 病毒可殘留於接觸過的物體上，因此量測方式應該採取非接觸式的體溫量測。而由找到的資料發現，透過 Arduino 與紅外線體溫量測元件模組可以完成非接觸式的體溫量測。

將所獲得的學生身份資訊加上所量測到的體溫進行資料的整合，資料整合完成後再上傳到網路平台，並輸出體溫監控表單，提供導師與學校行政人員監控學生的體溫狀況。若完成此裝置，將可解決上述體溫監控流程的缺點，達到提高效率、省人力、減少資訊誤植機率的效果，大大降低學校在學生體溫監控上的工作量，而我們稱此裝置為『紅外線體溫量測智慧登錄系統』，但要完成『紅外線體溫量測智慧登錄系統』的開發，須完成以下研究目的：

- (一)、學生學號身份辨識
- (二)、紅外線體溫量測
- (三)、紅外線體溫量測智慧登錄系統整合與資料處理

四、文獻回顧

疫情期間，量測學生體溫並記錄已是各級學校每天必要之工作，因此各個學校也都開發出各自的體溫量測系統，大部分的系統都是用使用紅外線體溫量測原理進行非接觸式的體溫量測，而在學生身份確認方面，便有很多方法，由我們所找到資料中發現，目前自動辨識學生身份的方式有：

- 1.使用個人磁扣[1]
- 2.使用個人證件條碼(如身份證、學生證、健保卡等)[2]
- 3.人臉辨識[3]

以上方式對國中生來說皆為較不可行之方法，使用個人磁扣將增加學校經費開銷，且磁扣上無記名，易搞混或弄丟；使用個人證件條碼，因需要滿 14 歲才能辦理身份證，因此未滿 14 歲之國中生將無法使用身份證辨識，且使用個人證件條碼可能會有冒名現象發生；而人臉辨識身份，需有大量個人照片提供 AI 訓練辨識模型，此前置作業非常繁瑣複雜，且會有肖像權的問題，因此較不適合運用於國中校園。

在國中的校園中，學生都必須穿著學校所規定的制服、運動服及學校外套，且制服、運動服及學校外套都必須繡上學生的學號，所以較不會有冒名的情況發生，可以較準確地確認學生身份，而學號為學生入學時學校所提供，因此校方已有學生學號的相關資料，無須重新建構身份資料，根據以上的優點，本研究將以學號作為學生身份辨識的依據。

五、研究原理

學號為一串數字所組成，因此我們就開始思考要如何獲得學生身上的學號資訊並進行身份辨識，由我們所找到的資料中發現，現在智慧停車場都採用車牌自動辨識系統記錄進出車輛。車牌結構為英文字母加上數字的組合，此結構與學號類似，因此我們將採用此技術來獲得學生身上的學號資訊。我們也發現最新的發票兌獎 App 可以自動辨識傳統發票上的統一發票號碼，並進行兌獎。因此我們開始思考是否可以將車牌自動辨識的技術與發票兌獎 App 的辨識兌獎技術結合，來獲取學生學號以辨識學生身份。而這兩項技術分別使用了 OpenCV 與 OCR(Optical Character Recognition，光學字元辨識)等相關技術。

溫度量測方面，由找到的資料發現，可以透過 Arduino 與 GY-MCU90615 紅外線體溫量測元件模組來完成非接觸式的體溫量測。因此以下將針對 OpenCV、OCR 以及 GY-MCU90615 紅外線體溫量測元件模組進行簡單說明與介紹：

(一)、OpenCV

OpenCV 的全名為 Open Source Computer Vision Library，是一個跨平台的電腦視覺庫。OpenCV 最初是由英特爾公司發起並參與開發，以 BSD 授權條款授權發行，可以在商業和研究領域中免費使用。OpenCV 可用於開發即時圖像處理、電腦視覺以及圖型識別程式等[4]。本研究將使用 Python 程式來執行 OpenCV 進行即時影像處理，判斷學號位置，並抓取學號位置之圖檔，以利後續的文字辨識。

Python 安裝 OpenCV 套件指令為：`pip install opencv-python`

(二)、OCR

OCR (Optical Character Recognition) 為光學字元辨識，是一個可將圖檔中的文字資料進行分析辨識處理，取得文字及版面資訊的技術，也就是將圖片或是影片中的文字數位化，常用於車牌辨識、字跡辨識、智慧鏡頭、發票兌獎 App 等。目前可用在 Python 的 OCR 套件有 pytesseract、easyocr、paddleocr 等，而由找到的資料中發現，easyocr 對數字辨識有較佳的效果[6]，且套件檔案較小，因此本研究將使用 easyocr 套件來辨識學生學號。

Python 安裝 easyocr 套件指令為：`pip install easyocr`

(三)、GY-MCU90615 紅外線體溫量測元件

物體溫度在絕對零度 (-273.15°C) 以上，就會發射出電磁輻射，在室溫之下，因為溫度低，物體發出的電磁輻射波長很長，落在紅外線範圍內，人眼看不到。所以只要能偵測紅外線，就能了解物體溫度。本研究使用 GY-MCU90615(圖 1.1)來偵測人體額頭所發出的紅外線。

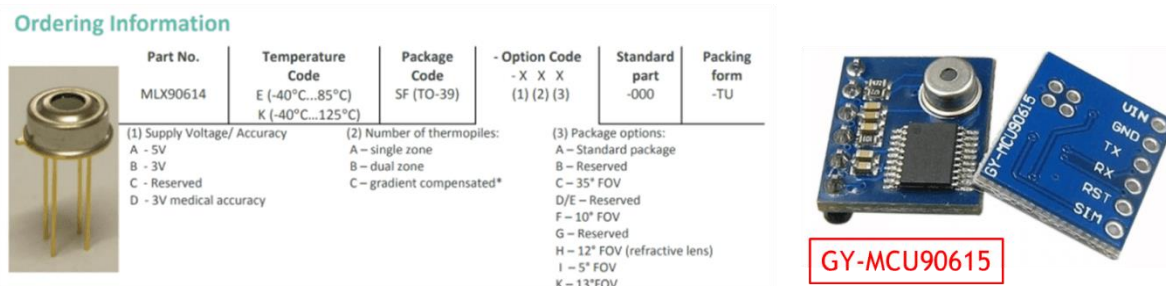






圖 1.1 GY-MCU90615 紅外線體溫量測元件




貳、研究設備及器材

一、紅外線體溫量測智慧登錄系統硬體設備

			
Arduino UNO	GY-MCU90615 紅外線體溫量測元件模組	RGB-LED 環狀彩燈	筆記型電腦、視訊鏡頭

			
按鍵開關模組	LCD1602 液晶螢幕	透明壓克力球	模型架

二、紅外線體溫量測智慧登錄系統設計軟體

		
Arduino IDE	Python(3.9.7)	Thonny

參、研究過程與方法

本研究將使用 Python 執行 OpenCV 程式，來抓取即時影像中學生的學號資訊，執行 easyocr 判斷學號資訊並比對學生身份，當學生身份確認後，使用 Arduino 與 GY-MCU90615 量測學生體溫，將所量測到的體溫回傳至 Python 程式，透過程式將收到的數據進行整合，最後將學生的個人資訊(班級、姓名、座號等)與體溫數值進行登錄，並上傳至網路平台，產生出體溫監控表單。研究架構流程圖如下：

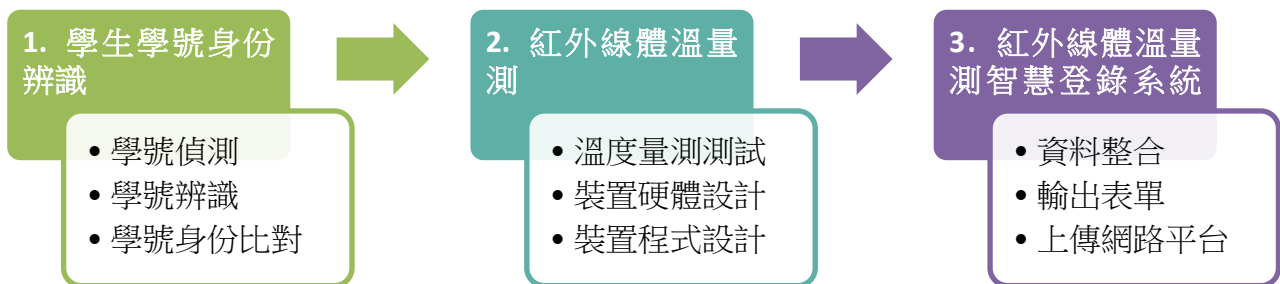


圖 3.1 紅外線體溫量測智慧登錄系統研究架構流程圖

一、學生學號身份辨識

實驗一、學號偵測

此實驗中，我們用運 OpenCV 取得視訊鏡頭影像，進行學號偵測，當偵測到學號後，框列出學號所在位置，並擷取所框列之影像，如圖 3.2。我們使用 OpenCV 的 Harr 串聯分類器來進行學號偵測，所使用的 Harr 串聯分類器為：haarcascade_russian_plate_number.xml，可偵測車牌或是數字串。

程式碼如下：

```
import cv2
plateCascade=cv2.CascadeClassifier("haarcascade_russian_plate_number.xml")
minArea = 500
cap =cv2.VideoCapture(0)
while True:
    _ , img = cap.read()
    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    numberPlates = plateCascade .detectMultiScale(imgGray, 1.05, 2)
    for (x, y, w, h) in numberPlates:
        area = w*h
        if area > minArea:
            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 4)
```

```

cv2.putText(img, "Student ID", (x+25, y-10),
            cv2.FONT_HERSHEY_COMPLEX, 1, (0, 150, 255), 2)
imgRoi = img[y:y+h, x:x+w]
cv2.imshow("ROI", imgRoi)
cv2.imshow("Result", img)

```

程式輸出結果：

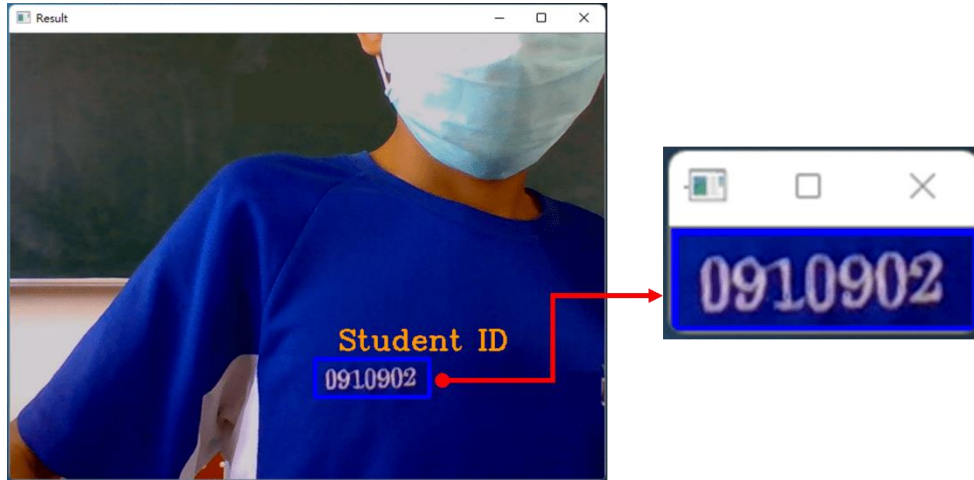


圖 3.2 學號偵測

實驗二、學號辨識

此實驗將上述實驗中的所擷取的學號影像，由 easyocr 套件進行影像中的文字辨識，並輸出結果，如圖 3.3。

程式碼如下：

```

import easyocr
if cv2.waitKey(1) & 0xFF == ord('s'):
    reader = easyocr.Reader(['en'], gpu=False)
    result = reader.readtext(imgRoi, detail = 0)
    print(result)

```

程式輸出結果：

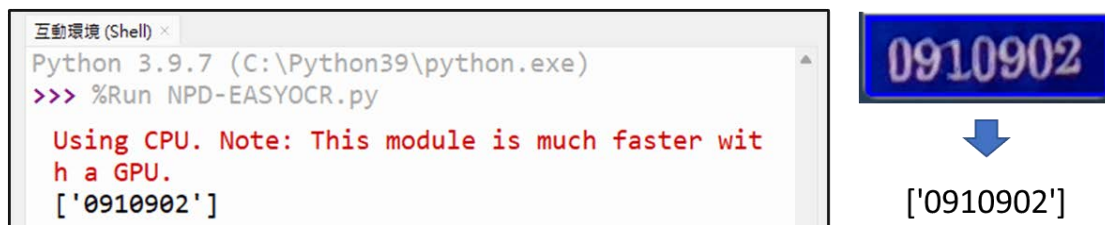


圖 3.3 學號影像 OCR

實驗三、學號身份比對

將所辨識出來的學號與學生資料進行比對，學生資料格式依序為學號、姓名、班級、座號，如圖 3.5。依學號進行比對，如果比對成功，依序輸出學號、班級、座號、姓名；如果比

對失敗，輸出文字：try again。

程式碼如下：

```
import pandas as pd
STD = pd.read_excel("student-data.xlsx")
SDF = pd.DataFrame(STD)
SDF.columns =
["student-ID", "student-name", "classroom", "number"]
SDF["student-ID"]=SDF["student-ID"].apply(str)
nSDF=[]
SID=[]
name=[]
a=[]
if cv2.waitKey(1) & 0xFF ==ord('s'):
    reader = easyocr.Reader(['en'],gpu=False)
    result = reader.readtext(imgRoi,detail = 0)
    print(result)
    SID=str(result)[3:9]
    print(SID)
    nSDF=SDF["student-ID"].isin([SID])
    a=SDF[nSDF].values
    print(a)
    if SDF[nSDF].empty:
        print('try again')
    else:
        name=a[0,1]
        classroom=a[0,2]
        seatnumber=a[0,3]
        print(SID,classroom,seatnumber,name)
```

程式輸出：

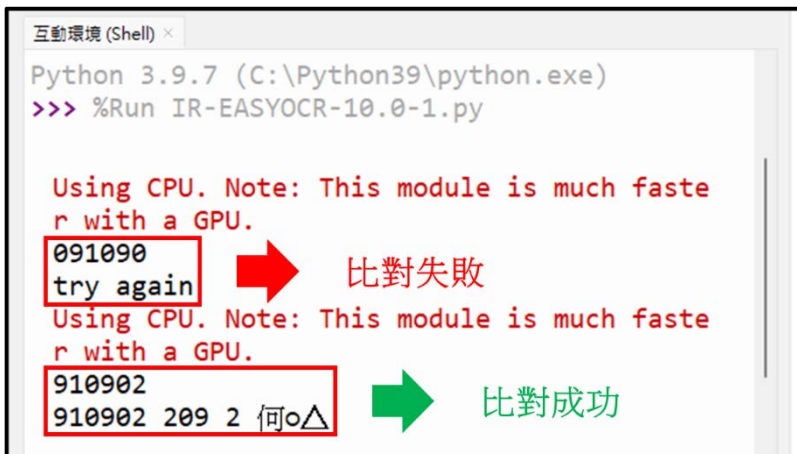


圖 3.4 學號身份比對

	A	B	C	D
1	學號	姓名	班級	座號
2	910101	王○△	201	1
3	910106	馬○△	201	6
4	920122	陳○△	201	22
5	910902	何○△	209	2
6	920128	羅○△	201	28
7	920217	夏○△	202	17
8	910301	王○△	203	1
9	910304	李○△	203	4
10	920318	吳○△	203	18
11	920326	黃○△	203	26
12	910403	李○△	204	3

圖 3.5 學號身份比對資料

二、紅外線體溫量測

實驗四、溫度量測測試

本實驗將運用 Arduino 程式控制 GY-MCU90615 進行體溫的量測，將量測結果與市售額溫槍的量測結果進行比對。硬體線路連接圖如圖 3.6。

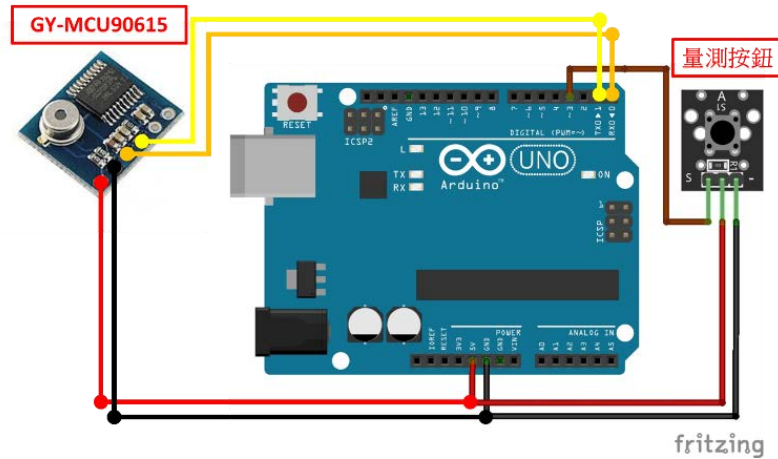


圖 3.6 溫度量測測試實驗硬體連接圖

Arduino 程式碼如下：

```
unsigned char Re_buf[11], counter = 0;
unsigned char sign = 0;
float TO = 0, TA = 0;
#include <Wire.h>
int buttonState = 0;
void setup() {
  Serial.begin(115200);
  delay(1);
  Serial.write(0XA5);
  Serial.write(0X45);
  Serial.write(0XEA);
  pinMode(13, OUTPUT);
  pinMode(3, INPUT);
}
void loop()
{
  buttonState = digitalRead(3);
  if (buttonState == LOW)
  {
    unsigned char i = 0, sum = 0;
    if (sign)
    {
      sign = 0;
      for (i = 0; i < 8; i++)
        sum += Re_buf[i];
    }
  }
}
```

```

if (sum == Re_buf[i] )
{
  TO = (float)(Re_buf[4] << 8 | Re_buf[5]) / 100;
  Serial.print("TO:");
  Serial.println(TO);
  delay(3000);
}
}
}
void serialEvent() {
  while (Serial.available()) {
    Re_buf[counter] = (unsigned char)Serial.read();
    if (counter == 0 && Re_buf[0] != 0x5A) return;
    counter++;
    if (counter == 9)
    {
      counter = 0;
      sign = 1;
    }
  }
}
}

```

程式輸出結果：

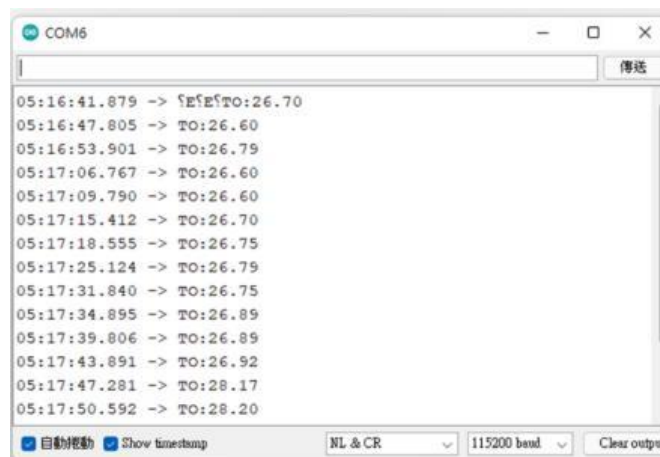


圖 3.7 溫度量測結果

同時使用市售額溫槍與 GY-MCU90615 進行 30 次的額溫量測，並比對量測結果，比對結果如圖 3.8，由圖 3.8 可以看出 GY-MCU90615 所量測出來的額溫是低於市售額溫槍的所量測的額溫，但有相似的變化趨勢，只須將 GY-MCU90615 的量測結果進行校正，就可以獲得較準確的溫度。將兩者資料畫成散佈圖並畫出趨勢線，如圖 3.9，並將兩者資料進行線性回歸處理，如表 3.1，而趨勢線之方程式為： $y = 0.3185x + 26.518$ ，因此校正方程式為：

$$\text{量測額溫} = 0.3185 \times (\text{GYMCU90615 量測結果}) + 26.518$$

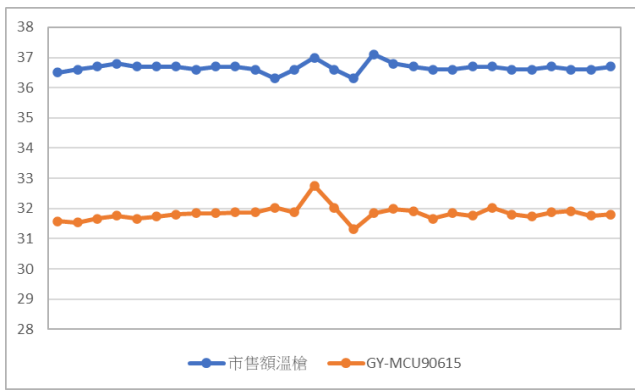


圖 3.8 量測溫度比對

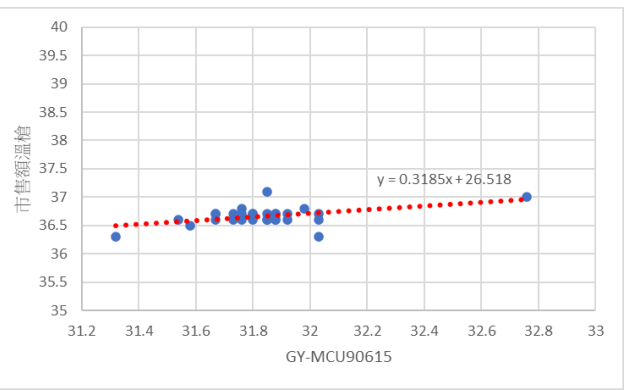


圖 3.9 溫度量測迴歸圖

表 3.1 溫度校正線性回歸資料表

	係數
截距	26.518
GY-MCU90615(x 值)	0.3185

實驗五、裝置硬體設計

本實驗將進行裝置的硬體設計，為了使量測者可以清楚知道自己量測的體溫狀況，我們將上個實驗的量測裝置加裝 LCD 螢幕以及環狀彩燈，如圖 3.10。為了方便學生量測額溫，因此將 GY-MCU90615 與環狀彩燈使用模型架架高，並安裝於透明壓克力球內，如圖 3.11。

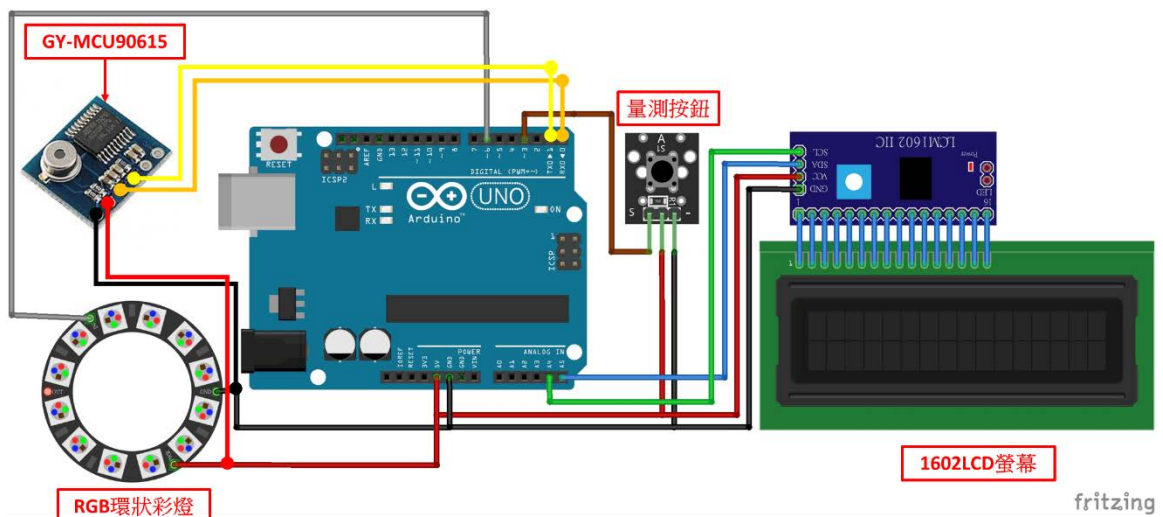


圖 3.10 裝置硬體設計連接圖

量測完成時，LCD 螢幕顯示量測者的溫度，當體溫過高時，環狀彩燈亮紅光，表示量測者體溫異常；體溫正常時亮綠光，表示體溫正常。

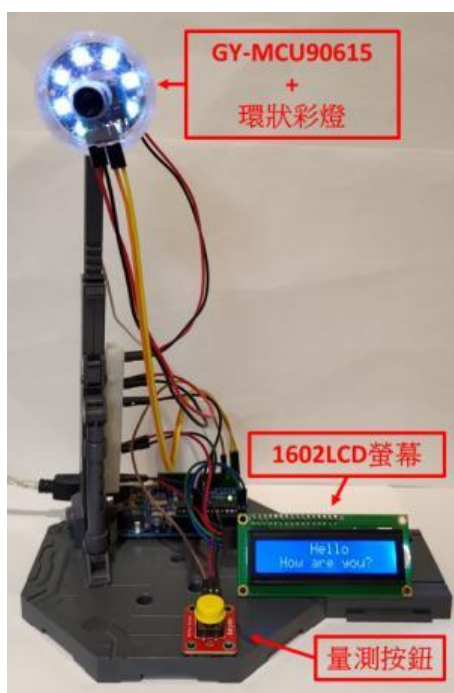


圖 3.11 裝置硬體實際連接圖

實驗六、裝置程式設計

本實驗將說明 Arduino 程式設計，當量測者額頭靠近 GY-MCU90615 時，按下量測按鈕就可以量測其額溫，量測完成後，LCD 螢幕將顯示所量測到的溫度；未按下量測按鈕時，LCD 螢幕將顯示：Hello How are you?，如圖 3.12。我們將 37.5°C 作為體溫過高與正常的判斷標準，當所量測到的體溫 $\geq 37.5^{\circ}\text{C}$ 時，環狀彩燈亮起紅色色光以做警示；若所量測到的體溫 $< 37.5^{\circ}\text{C}$ 時，環彩燈亮起綠色色光；未進行體溫量測時環狀彩燈亮起白色色光，如圖 3.13。

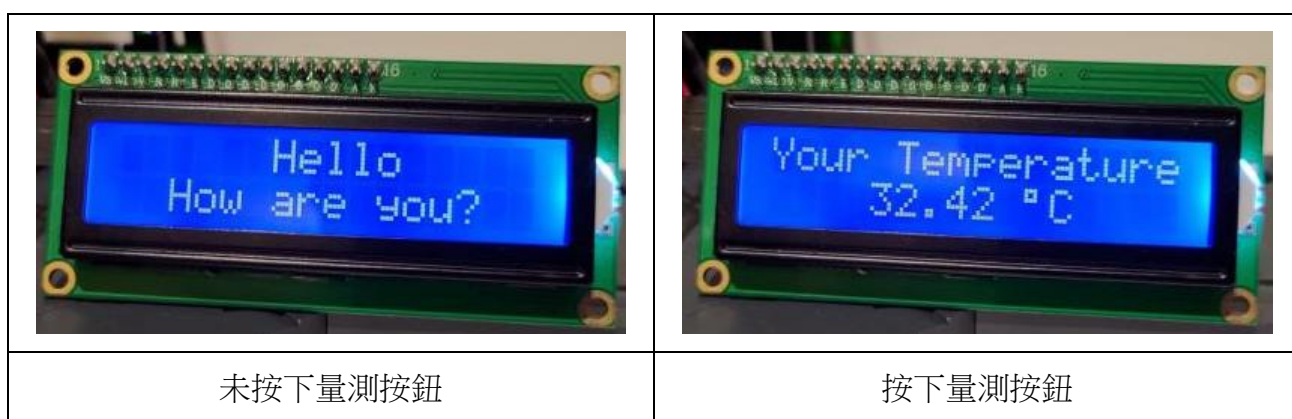


圖 3.12 LDC 螢幕顯示



圖 3.13 環狀彩燈點亮圖

Arduino 程式碼如下：

```

unsigned char Re_buf[11], counter = 0;
unsigned char sign = 0;
float T0 = 0, TA = 0;
#include <LiquidCrystal_PCF8574.h>
#include <Wire.h>
#include <Adafruit_NeoPixel.h>
#define PIN 6
#define NUMPIXELS 8
Adafruit_NeoPixel pixels=Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800);
int brightness=255;
int c;
LiquidCrystal_PCF8574 lcd(0x3F);
int buttonState = 0;
void setup() {
  Serial.begin(115200);
  delay(1000);
  Serial.write(0XA5);
  Serial.write(0X45);
  Serial.write(0XEA);
  lcd.begin(16, 2); // 初始化 LCD
  lcd.setBacklight(255);
  lcd.clear();
  pinMode(13, OUTPUT);
  pinMode(3, INPUT);
  pixels.begin();
  pixels.setBrightness(brightness);
}
void loop()
{
  buttonState = digitalRead(3);
  if (buttonState == LOW)
  {
    unsigned char i = 0, sum = 0;
    if (sign)
  
```



```

{
  sign = 0;
  for (i = 0; i < 8; i++)
    sum += Re_buf[i];
  if (sum == Re_buf[i] )
  {
    T0 = (float)(Re_buf[4] << 8 | Re_buf[5]) / 100;
    //Serial.print("T0:");
    Serial.println(T0*0.3185+26.518);
    //Serial.println(T0);

    if(T0*0.3185+26.518 >= 37.5)
    {
      for(c=0;c<8;c++)
      {
        pixels.setPixelColor(c,pixels.Color(50, 0, 0));
        pixels.show( );
      }
    }
    else if(T0*0.3185+26.518 < 37.5)
    {
      for(c=0;c<8;c++)
      {
        pixels.setPixelColor(c,pixels.Color(0, 50, 0));
        pixels.show( );
      }
    }
    lcd.clear();
    lcd.setCursor(0, 0); //設定游標位置 (字,行)
    lcd.print("Your Temperature");
    lcd.setCursor(4, 1);
    lcd.print(T0*0.3185+26.518);
    lcd.setCursor(10, 1);
    lcd.print((char)223); //用特殊字元顯示符號的"度"
    lcd.setCursor(11, 1);
    lcd.print("C");
    delay(2000);
  }
}

```



```

}

else
{
  lcd.setCursor(0, 0);
  lcd.print("      Hello      ");
  lcd.setCursor(0, 1);
  lcd.print(" How are you? ");
  for(c=0;c<8;c++)
  {

```

```

        pixels.setPixelColor(c,pixels.Color(10, 10, 10));
        pixels.show( );
    }
}
void serialEvent() {
    while (Serial.available()) {
        Re_buf[counter] = (unsigned char)Serial.read();
        if (counter == 0 && Re_buf[0] != 0x5A) return;
        counter++;
        if (counter == 9)
        {
            counter = 0;
            sign = 1;
        }
    }
}
}

```



三、紅外線體溫量測智慧登錄系統整合與資料處理

實驗七、資料整合

本實驗將 GY-MCU90615 所測到的溫度結果由電腦連接埠輸入 python，再與當下時間(包含日期與時間)以及所辨識出的學號資訊(包含班級、姓名、座號)結合後輸出，如圖 3.14。

所使用的 python 套件為：pySerial。Python 安裝 pySerial 套件指令為：pip install pySerial

程式碼如下：

```

import serial
import datetime
Date = datetime.datetime.now()
date = Date.strftime('%Y/%m/%d')
COM_PORT = 'COM6'    # 指定通訊埠名稱
BAUD_RATES = 115200  # 設定傳輸速率
ser = serial.Serial(COM_PORT, BAUD_RATES)
if ser.in_waiting:# 若收到序列資料...
    data_raw = ser.readline() # 讀取一行
    data = data_raw.decode("ISO-8859-1") # 用預設的 UTF-8 解碼
    STP=str(data)
    if a==[]:
        print('Start')
        #print(a)
    else:
        #print(data_raw)
        print(data)
        Time = datetime.datetime.now()
        time = Time.strftime('%H:%M:%S')
        print(date,time,SID,classroom,seatnumber,name,STP)

```


程式輸出：



圖 3.14 學生資料與溫度資料整合

實驗八、輸出表單

接著將上述的實驗結果存成 CSV 檔，以便後續查看與追蹤，採用量測一筆資料就記錄一次的方式，來完成學生的體溫監控表單，如圖 3.15。我們將資料登錄欄位順序設計為：日期、時間、學號、班級、座號、姓名、溫度。

實驗程式碼如下：

```
import csv
import pandas as pd
df = pd.DataFrame
([[ '日期', '時間', '學號', '班級', '座號', '姓名', '溫度' ]],
 columns = [ 'Date', 'Time', 'Student ID', 'Classroom', 'Seat number', 'Name',
 'Student Temperature' ])
df.to_csv("data-test2.csv", index = False, encoding="utf-8-sig")
if ser.in_waiting:# 若收到序列資料...
    data_raw = ser.readline() # 讀取一行
    data = data_raw.decode("ISO-8859-1") # 用預設的 UTF-8 解碼
    STP=str(data)
    if a==[]:
        print('Start')
        #print(a)
    else:
        #print(data_raw)
        print(data)
        Time = datetime.datetime.now()
        time = Time.strftime('%H:%M:%S')
        print(date,time,SID,classroom,seatnumber,name,STP)
        df2 = pd.DataFrame([[date,time,SID,classroom,seatnumber,
            name,STP]], columns = [ 'Date', 'Time', 'Student ID',
            'Classroom', 'Seat number', 'Name', 'Student Temperature' ])
        df2.to_csv("data-test2.csv", mode = 'a', header = False,
            index = False)
```

程式表單輸出：

	A	B	C	D	E	F	G
1	Date	Time	Student ID	Classroom	Seat number	Name	Student Temperature
2	日期	時間	學號	班級	座號	姓名	溫度
3	2022/3/18	12:28:51	920420	204	20	吳○△	37.80
4	2022/3/18	12:32:08	920125	201	25	賴○△	37.80
5	2022/3/18	12:32:41	920122	201	22	陳○△	38.07
6	2022/3/18	12:34:29	920128	201	28	羅○△	37.97
7	2022/3/18	12:35:14	910106	201	6	馬○△	38.36
8	2022/3/18	12:35:47	920217	202	17	夏○△	37.88
9	2022/3/18	12:36:07	920821	208	21	洪○△	38.10
10	2022/3/18	12:40:16	910503	205	3	王○△	37.85

圖 3.15 體溫監控表單

實驗九、上傳網路平台

因體溫監控表單儲存於量測電腦中，教師無法進行遠端即時監控。因此使用 `gspreed` 與 `oauth2client` 套件並搭配 `google sheets API`，將體溫監控表單即時上傳至教師的 `google sheets`，達到可以遠端監控的目的，如圖 3.16。

實驗程式碼如下：

```
import gspread
from google.oauth2.service_account import Credentials
#google sheet 授權
scopes = ['https://www.googleapis.com/auth/spreadsheets',
          'https://www.googleapis.com/auth/drive']
credentials = Credentials.from_service_account_file(
    'robust-cooler-344322-b64d91cce03b.json',scopes=scopes)
gc = gspread.authorize(credentials)
spreadsheet_key='1j0265rGg2769yHv1AWKcUI5kLrukcg3vc1-foIBQFy8'
sheet=gc.open_by_key(spreadsheet_key).worksheet("工作表 1")
if ser.in_waiting:# 若收到序列資料...
    data_raw = ser.readline() # 讀取一行
    data = data_raw.decode("ISO-8859-1") # 用預設的 UTF-8 解碼
    STP=str(data)
    if a==[]:
        print('Start')
        #print(a)
    else:
        #print(data_raw)
        print(data)
        Time = datetime.datetime.now()
        time = Time.strftime('%H:%M:%S')
```

```
print(date,time,SID,classroom,seatnumber,name,STP)
upload=[date,time,SID,classroom,seatnumber,name,STP]
sheet.append_row(upload)
```

程式輸出結果：

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	日期	時間	學號	班級	座號	姓名	溫度						
2	2022/03/18	12:28:51	920420	204	20	吳○△	37.8						
3	2022/03/18	12:32:08	920125	201	25	賴○△	37.8						
4	2022/03/18	12:32:41	920122	201	22	陳○△	38.07						
5	2022/03/18	12:34:29	920128	201	28	羅○△	37.97						
6	2022/03/18	12:35:14	910106	201	6	馬○△	38.36						
7	2022/03/18	12:35:47	920217	202	17	夏○△	37.88						
8	2022/03/18	12:36:07	920821	208	21	洪○△	38.10						
9	2022/03/18	12:40:16	910503	205	3	王○△	37.85						
10	2022/03/18	12:40:50	910510	205	10	陳○△	37.67						
11	2022/03/18	12:41:26	910909	209	9	黃○△	37.76						
12	2022/03/18	12:43:45	910902	209	2	何○△	37.88						
13	2022/03/18	12:44:19	920722	207	22	廖○△	37.97						
14	2022/03/18	12:54:31	910705	207	5	邱○△	35.92						
15	2022/03/18	13:52:54	876431	210	31	郭○△	36.29						
16	2022/03/18	14:19:29	920420	204	20	吳○△	36.07						
17	2022/03/18	14:19:56	920426	204	26	陳○△	36.07						
18	2022/03/19	10:11:30	973232	304	38	郭○△	34.64						
19	2022/03/21	09:51:48	920722	207	22	廖○△	34.29						

圖 3.16 google sheets 上的體溫監控表單

四、學號偵測辨識優化



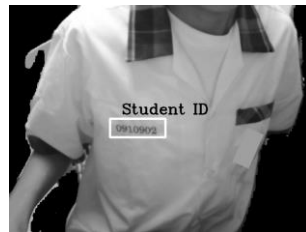
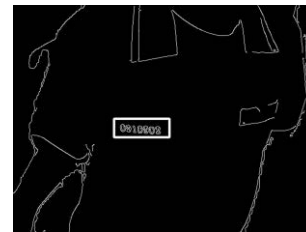
實驗十、學號偵測辨識優化

在開發過程中，發現本校制服的學號偵測辨識效果不好，偵測時間長且辨識容易失敗，因此設計此實驗，希望能找出方法來解決制服不易辨識的問題，並優化學號偵測辨識系統。學號偵測辨識系統由學號偵測(實驗一)與學號辨識(實驗二)所組成，而在多次實驗後發現，鏡頭畫面內的背景景物與景物色彩似乎會影像學號的偵測，導致 Harr 串聯分類器無法快速偵測學號位置，因此我們將鏡頭畫面先進行影像處理，再由 Harr 串聯分類器進行學號位置偵測，我們的影像處理方法有影像去背景與邊緣檢測(Canny)。而在學號辨識方面，為了討論衣服底色是否影響辨識效果，將實驗一所框出的學號先進行色彩處理，再由 easyocr 進行辨識，我們學號色彩處理方式有黑白辨識與彩色辨識(維持原畫面)。學號色彩處理辨識有：彩色辨識、黑白變色，如表 3.2。鏡頭畫面影像處理偵測有：全彩偵測、灰階偵測、(灰階+去背)偵測、(去背+Canny)偵測，如表 3.3。

表 3.2 學號色彩處理辨識

彩色辨識		黑白辨識	

表 3.3 鏡頭畫面影像處理偵測

全彩偵測	灰階偵測	灰階+去背偵測	去背+Canny 偵測
			

將上述的偵測方式與辨識方式結合，將有全彩偵測+彩色辨識、灰階偵測+彩色辨識、(灰階+去背)偵測+彩色辨識、(去背+Canny)偵測+彩色辨識、全彩偵測+黑白辨識、灰階偵測+黑白辨識、+黑白辨識、(去背+Canny)偵測+黑白辨識，共八種學號偵測辨識方法，我們將使用這八種方法進行三種校服(運動服、外套、制服)的學號偵測辨識，透過多次偵測辨識後得到辨識成功率與辨識成功時間，找出最佳的學號偵測辨識方法，初步的實驗結果如表 3.4。

表 3.4 學號偵測辨識優化初步實驗結果

	運動服(彩色)	運動服(黑白)	外套(彩色)	外套(黑白)	制服(彩色)	制服(黑白)
全彩偵測	成功辨識	成功辨識	成功辨識	成功辨識	辨識失敗	辨識失敗
灰階偵測	成功辨識	成功辨識	成功辨識	成功辨識	辨識失敗	辨識失敗
灰階+去背	成功辨識	成功辨識	成功辨識	成功辨識	辨識失敗	辨識失敗
去背+Canny	成功辨識	成功辨識	成功辨識	成功辨識	成功辨識	成功辨識

肆、研究結果

一、紅外線體溫量測智慧登錄系統

由前面的實驗過程，我們已經可以透過 OpenCV 中的 Harr 串聯分類器偵測影像中學號的位置，並進行學號影像的框列。而框列出來的學號影像可透過 easyocr 進行學號辨識，將圖片學號轉成字串學號，將字串學號與學生資料比對。比對成功後，進行體溫的量測，接著將學生資料與量測到的體溫整合，最後將整合好的資料輸出成體溫監控表單，並同時上傳師長的雲端 google sheets，達到可以遠端監控的目的，因此我們已經成功開發出『紅外線體溫量測智慧登錄系統』。

使用『紅外線體溫量測智慧登錄系統』進行體溫監控時，只須先將紅外線體溫量測裝置與電腦連接，如圖 4.1，連接後開啟 python 程式就可以開始量測學生的體溫，而同學可以自行操作，無須額外派人力幫忙量測與記錄。

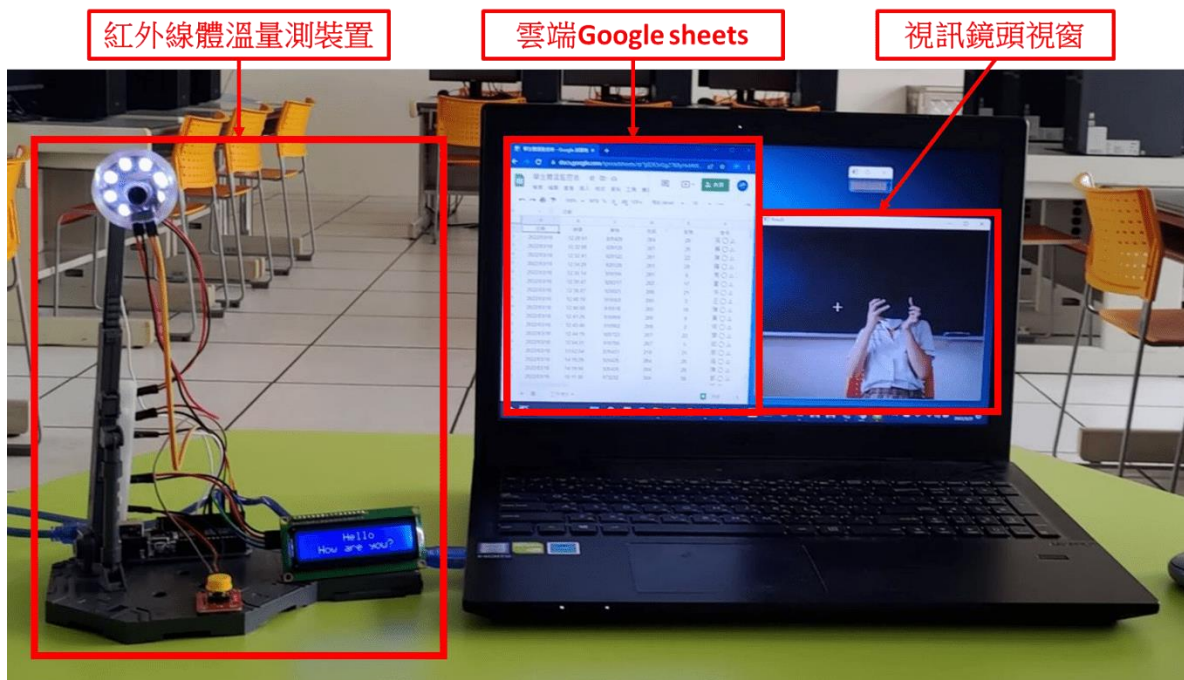


圖 4.1 紅外線體溫量測裝置連接電腦

使用『紅外線體溫量測智慧登錄系統』時，因為量測只需要三個步驟，所以降低了量測與記錄的時間，根據我們多次量測所得到的結果，平均一位同學量測的時間約為 10 秒，若是較熟悉量測步驟的同學，量測時間可以控制在 5 秒內，以一個班級三十位同學來計算的話，整班的量測時間約可以控制在 10 分鐘內，也就是一個下課就可完成整班的體溫量測並登錄。

量測完成後，程式可自行輸出體溫監控表單，同時也直接上傳至師長的雲端 google sheets，省去體溫監控表單手寫登錄，以及由學生幹部繳交至學校學務處的時間。因輸出的體溫監控表單為電子檔案，所以學務處也無須派人力將表單輸入成電子檔，目前我們已經將此系統運用於本校資優特殊班級的體溫監控上。

二、紅外線體溫量測智慧登錄系統偵測辨識優化結果

透過多次偵測辨識的實驗後得到各種偵測辨識方法的辨識成功率，實驗結果如下：

(一) 運動服偵測辨識實驗之結果

表 4.1 為運動服透過四種影像處理後的偵測畫面；圖 4.2 為運動服透過學號彩色辨識的成功率比較圖；圖 4.3 為運動服透過學號黑白辨識的成功率比較圖。

表 4.1 鏡頭畫面影像處理偵測-運動服

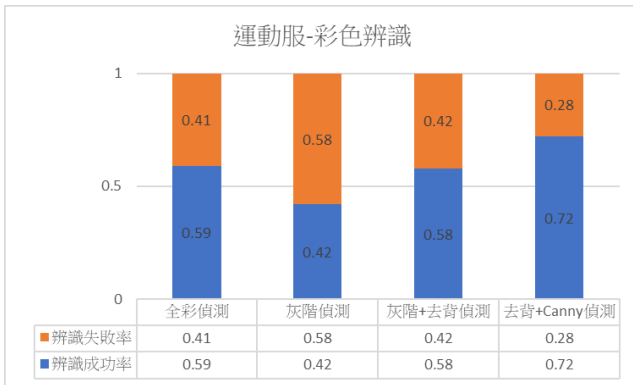
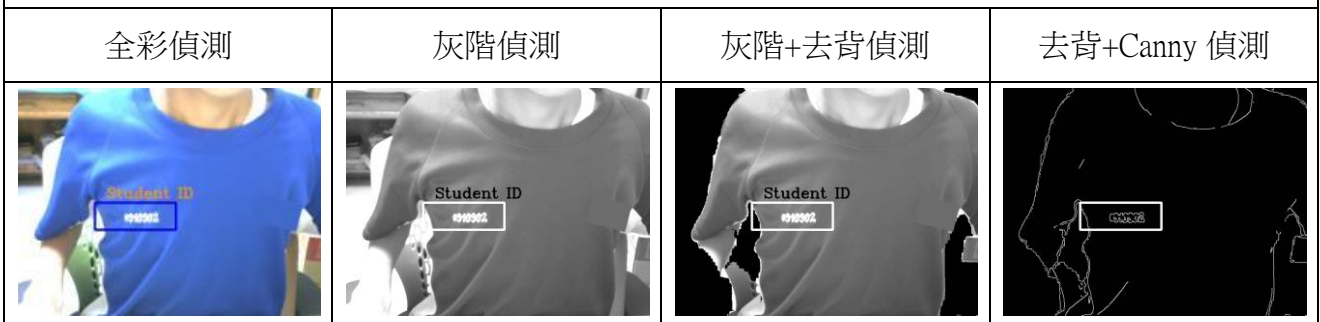


圖 4.2 運動服學號彩色辨識成功率比較圖

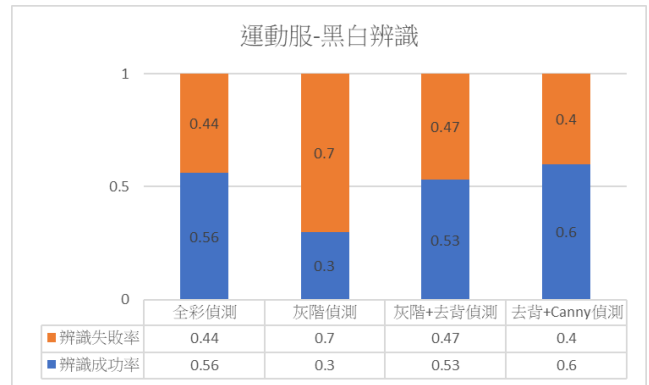


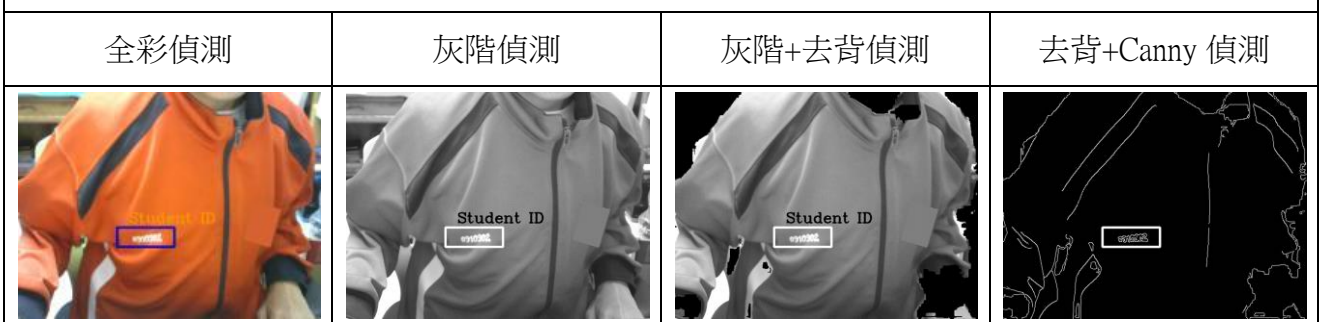
圖 4.3 運動服學號黑白辨識成功率比較圖

由圖 4.2 與圖 4.3 可以看出運動服不管是彩色辨識還是黑白辨識，在這四種偵測方式下都有類似的變化趨勢，都是(去背+Canny)偵測的成功率最高。而在學號色彩辨識的部分則是彩色辨識成功率優於黑白辨識的成功率。

(二) 外套偵測辨識實驗之結果

表 4.2 為外套透過四種影像處理後的偵測畫面；圖 4.4 為外套透過學號彩色辨識的成功率比較圖；圖 4.5 為外套透過學號黑白辨識的成功率比較圖。

表 4.2 鏡頭畫面影像處理偵測-外套



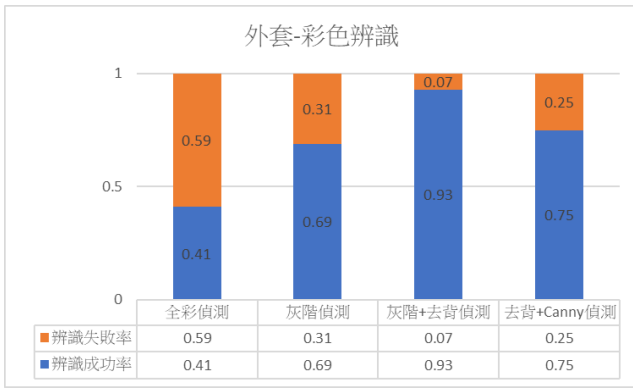


圖 4.4 外套學號彩色辨識成功率比較圖

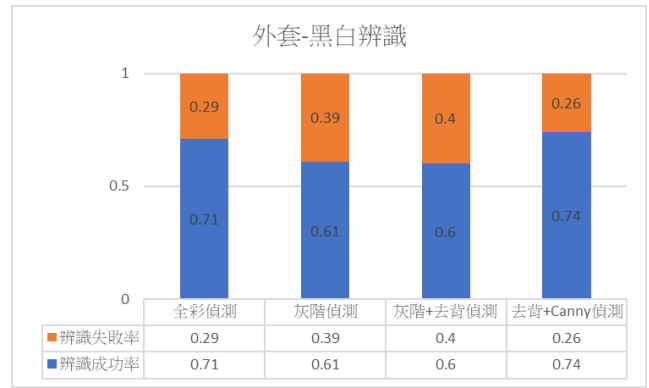

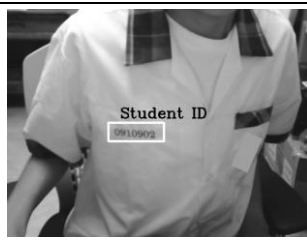

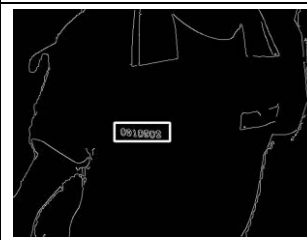


圖 4.5 外套學號黑白辨識成功率比較圖

由圖 4.4 與圖 4.5 可以看出外套在彩色辨識下的成功率幾乎都高於黑白辨識的成功率，且成功率幾乎都在 0.6 以上，其中(去背+Canny)偵測方式的成功率接近 0.75(彩色辨識：0.75；黑白辨識：0.74)。

(三) 制服偵測辨識實驗之結果

表 4.3 為制服透過四種影像處理後的偵測畫面；圖 4.6 為制服透過學號彩色辨識的成功率比較圖；圖 4.7 為制服透過學號黑白辨識的成功率比較圖。

表 4.3 鏡頭畫面影像處理偵測-制服				
全彩偵測	灰階偵測	灰階+去背偵測	去背+Canny 偵測	
				

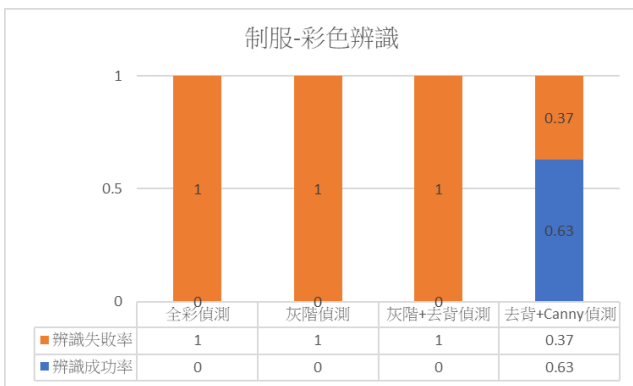


圖 4.6 制服學號彩色辨識成功率比較圖

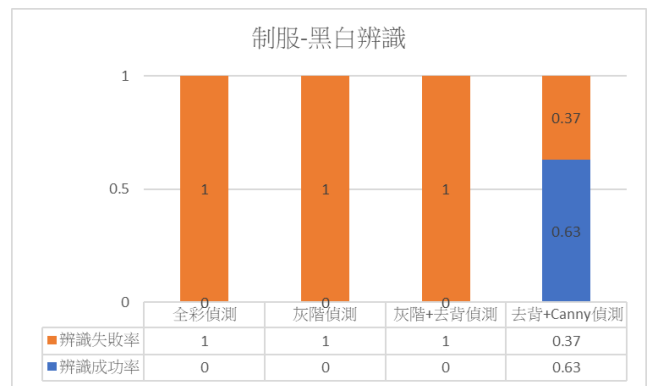


圖 4.7 制服學號黑白辨識成功率比較圖

由圖 4.6 與圖 4.7 可以看出制服不管在彩色辨識還是黑白辨識，除了(去背+Canny)偵測的

方式外，其餘的偵測方式都無法順利完成學號偵測辨識，因此制服使用(去背+Canny)偵測可解決制服學號無法辨識成功的問題，成功率也都高於 0.6(彩色辨識：0.63；黑白辨識：0.63)。

根據上述的實驗結果，(去背+Canny)偵測的方式可以解決制服無法順利辨識的問題，同時在運動服跟外套上都有不錯的辨識成功率，同時發現(去背+Canny)偵測加上彩色辨識的成功率較(去背+Canny)偵測加上黑白辨識的成功率來的更佳，如表 4.4。因此我們將選用(去背+Canny)偵測加上彩色辨識來當作紅外線體溫量測智慧登錄系統偵測辨識學號的方法。

	彩色辨識	黑白辨識
運動服	0.72	0.6
外套	0.75	0.74
制服	0.63	0.63

三、紅外線體溫量測智慧登錄系統結果顯示優化

經過多次的量測後，我們發現當量測結果顯示於 Thonny 的環境視窗(Shell)時，第一次使用或是不熟悉程式介面的同學，無法快速了解量測的結果，如圖 4.8。因此我們使用了 OpenCV 中的 cv2.putText 語法，將班級跟座號直接顯示於視訊鏡頭視窗，量測完溫度後，將溫度顯示於班級座號下方，讓量測結果顯示更加視覺化，如圖 4.9。

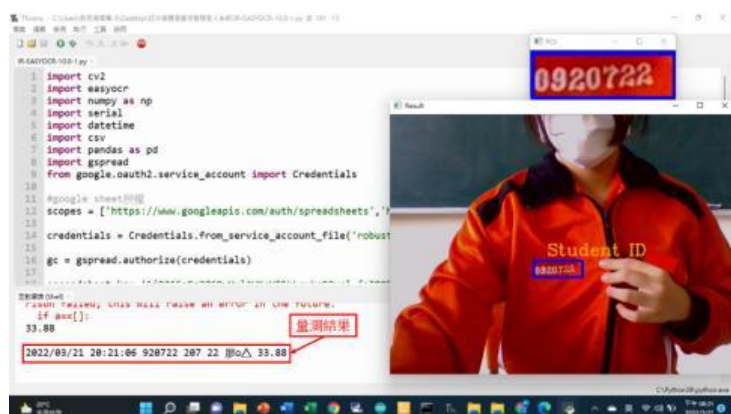


圖 4.8 Thonny 的環境視窗(Shell)顯示量測結果

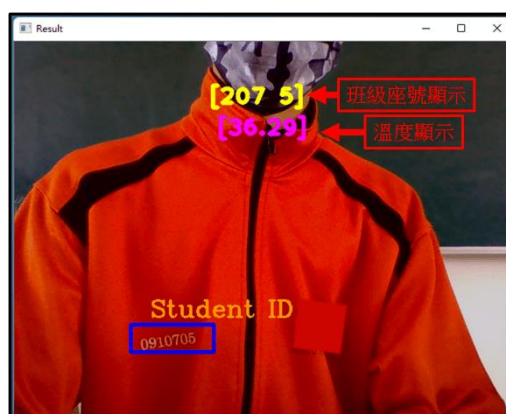
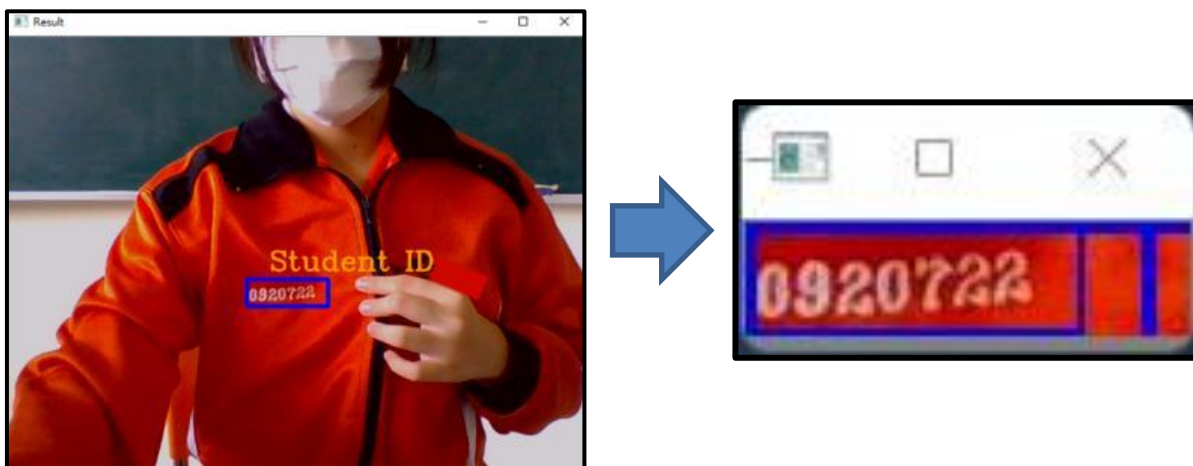


圖 4.9 結果顯示優化

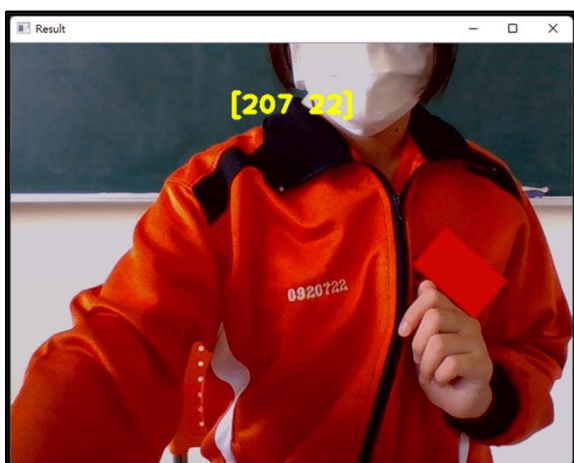
接著將說明『紅外線體溫量測智慧登錄系統』量測步驟與過程，並顯示優化後的結果，步驟說明如表 4.5。

表 4.5 『紅外線體溫量測智慧登錄系統』體溫量測步驟

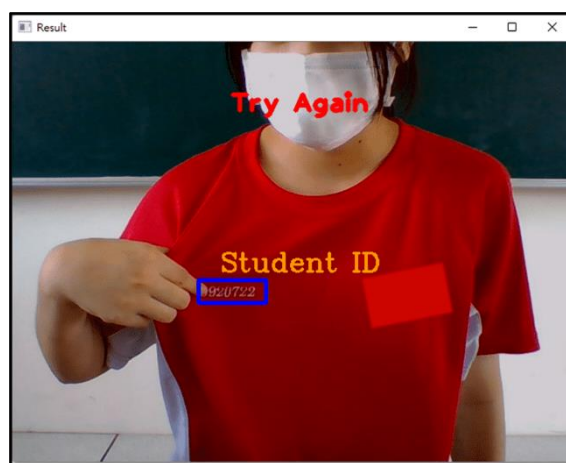
步驟一、讓程式偵測學號位置



步驟二、讓程式自動辨識學生身份（進行學號辨識與學生身份比對）

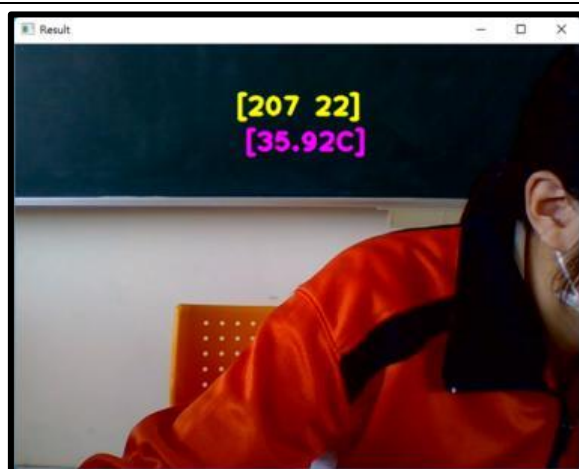


比對成功，顯示班級座號



比對失敗，顯示 Try Again（請重複步驟一）

步驟三、比對成功，按下量測按鈕，顯示量測溫度，完成量測並登錄。



伍、討論

一、學號偵測

經過多次量測發現，當穿著本校運動服與學校外套時有較好的偵測效果，穿著制服時，偵測效果較差，導致量測時間增加，如表 5.1。

表 5.1 不同校服的學號偵測情況		
		
外套	運動服	制服

推測原因為本校制服是白色的，所以較不易偵測出學號位置。而我們使用了兩個方法解決此問題，分別為使用學號偵測輔助工具以及偵測畫面影像處理。

1. 使用學號偵測輔助工具

我們所使用的 Harr 串聯分類器為：haarcascade_russian_plate_number.xml，可輕易辨識出影像中汽車車牌的位置，因此我們製作了黑色方框模擬汽車車牌樣式作為學號偵測的輔助工具，如圖 5.1。當穿著制服時，只須將黑框置於學號位置處，電腦程式就可以快速偵測到學號位置，如圖 5.2，以便後續的學號辨識，加快整體的量測流程。



圖 5.1 制服量測輔助方框

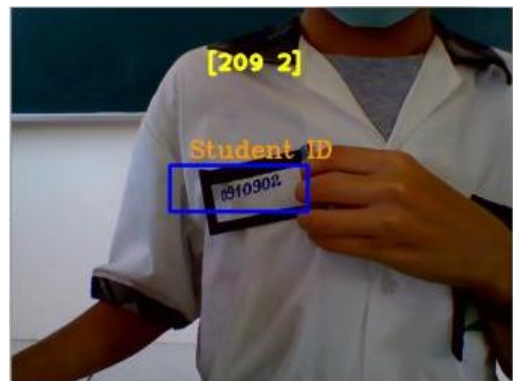


圖 5.2 使用輔助方框進行制服學號偵測

2. 偵測畫面影像處理

經過多次學號偵測後發現，偵測過程中程式有可能抓取到背景中的物體或是圖像，導致學號辨識失敗，如圖 5.3，所以我們將偵測畫面中的背景去除，如圖 5.4，降低背景的影響程度，同時也發現衣服上的皺褶與陰影似乎也會影像到學號的偵測結果，因此將去背影像再進行 Canny(邊緣檢測)處理，如圖 5.5，降低衣服上的皺褶與陰影的影響。



圖 5.3 學號偵測失敗

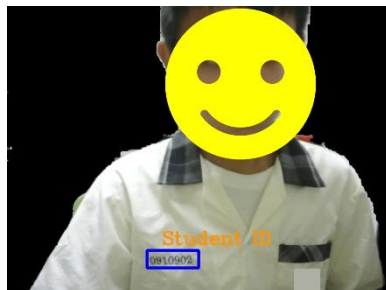


圖 5.4 去背學號偵測

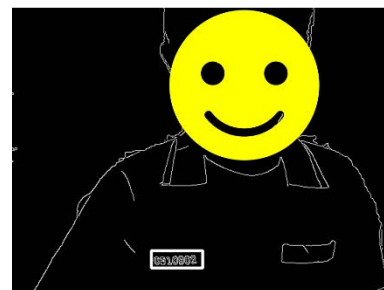


圖 5.5 Canny 學號偵測

只需將偵測畫面進行去背及 Canny 的影像處理，就可解決制服學號不易偵測的問題，並可以加快辨識成功的時間，且將辨識成功的平均時間控制約在 3.5 秒內，如表 5.2。

	運動服	外套	制服
辨識成功率	0.72	0.75	0.63
辨識成功平均時間	3.09(s)	2.87(s)	3.15(s)

而疫情期間應減少不必要的接觸，所以我們使用偵測畫面影像處理的方式來解決制服學號不易偵測成功的問題。

二、未來工作

目前我們已將『紅外線體溫量測智慧登錄系統』運用於本校資優特色班級上的體溫監控，未來希望能將此系統推廣至全校班級。為了讓此系統更方便被利用，未來研究工作目標如下：

1. 加裝聲音警示系統，當體溫過高時，可以發出聲音提醒量測者。
2. 了解 GPU 設定，並使用 GPU 執行學號辨識，使辨識過程更加快速，目前為使用 CPU 執行辨識，辨識時間約為 3.5 秒。
3. 將紅外線體溫量測裝置改成用藍芽與電腦連接，完成可在教室內移動的量測裝置。
4. 將此系統開發成手機 App，並結合上述的藍芽連線裝置，使此系統可以更加輕便的攜帶與移動，使校園內的防疫工作更加輕鬆。

陸、結論

1. 成功使用 Python OpenCV 套件，開發出可以自動偵測學號位置，並框列學號影像的偵測系統。
2. 成功使用 Python easyocr 套件進行學號影像辨識，並將影像學號轉成字串學號。
3. 成功使用 Python 程式將字串學號與學生身份進行比對。
4. 成功使用 Arduino 程式進行 GY-MCU90615 紅外線量測元件的測試，並與市售額溫槍進行量測結果校正。
5. 成功開發出紅外線體溫量測裝置的硬體設備。
6. 成功使用 Arduino 程式碼控制紅外線體溫量測裝置，並可快速進行體溫量測。
7. 成功使用 Python 程式將學生資料與量測體溫進行整合。
8. 成功使用 Python 程式將整合完成的資料，輸出成體溫監控表單。
9. 成功使用 Python 程式與師長的 google sheets API，將體溫監控表單上傳至師長的雲端 google sheets，以利進行即時遠端監控。
10. 成功使用(去背+Canny)偵測+彩色辨識的方式，解決制服學號不易辨識成功的問題，使紅外線體溫量測智慧登錄系統有更佳的辨識成功率(大於 0.6)與更快的辨識時間(約 3.5s)。
11. 成功開發出『紅外線體溫量測智慧登錄系統』，並已運用於本校資優特色班級，未來希望能將此系統推廣至全校班級。

柒、參考文獻

- [1] 劉智維(2020 年 3 月 20 日)・防疫動起來！高職師花千元打造額溫自動登錄系統・蘋果即時・取自 <https://tw.appledaily.com/local/20200320/3TYNKXAVXFIPBICVN6VKFZP2D4/>
- [2] 張立明(2020 年 3 月 20 日)・雲科大自動防疫檢測站 辨識測額溫消毒一條龍・亞太新聞網・取自 <https://www.atanews.net/?fun-net=66106>
- [3] 懷陽(2020 年 5 月 18 日)・基隆商工學生研發「結合 AI-量溫點名系統」科展獲優等・公民新聞・取自 <https://www.peopo.org/news/458142>
- [4] OpenCV・(2020 年 5 月 18 日)・維基百科・取自

<https://zh.wikipedia.org/w/index.php?title=OpenCV&oldid=68163631>

- [5] 施威銘研究室(2020) · 用 Python 學 AIoT 智慧連網 · 旗標科技股份有限公司
- [6] 郭英勝、鄭志宏、龔志銘、謝哲光 (2018) · 實用 Python 程式設計 · 碁峯資訊
- [7] 楊明豐(2018) · Arduino 物聯網最佳入門與應用：打造智慧家庭輕鬆學 · 碁峯資訊
- [8] Pratik Desai(2016) · Python x Arduino 物聯網整合開發實戰(CAVEDU 團隊,曾吉弘 譯) · 碁峯資訊
- [9] 國民中學自然與生活科技第三冊 · 第四章：光與顏色 · 翰林書局。
- [10] 國民中學自然與生活科技第三冊 · 第五章：熱的傳播 · 翰林書局。
- [11] 楊明豐(2020) · Arduino 物聯網最佳入門與應用：打造智慧家庭輕鬆學 · 台北市 · 碁峯資訊。
- [12] 黃建庭(2018) · 輕鬆玩 Arduino 程式設計與感測器入門 · 台北市 · 碁峯資訊。
- [13] 朱克剛(2021) · AIOT 與 OpenCV 實戰應用 · 台北市 · 碁峯資訊。
- [14] 李立宗(2020) · 科班出身的 AI 人必修課 OpenCV 影像處理 使用 Python · 台北市 · 深智數位股份有限公司。
- [15] 網路資源：easyocr：<https://github.com/JaidedAI/EasyOCR>

【評語】 032814

本作品利用 AI 影像辨識衣服上的學號，搭配紅外線體溫傳感器，開發一套可用於校園之智慧體溫量測登錄系統，作品對影像辨識之機制做最佳化的流程參數設計，構想有創意，惟溫度量測準確度誤差與穩定性評估、衣物皺褶時如何提高學號辨識成功率等方面，較缺乏系統性的科學分析與探討。鼓勵作者再接再厲，繼續精進，讓作品更完善。

作品簡報

紅外線體溫量測智慧登錄系統之開發與研究

科 別：生活與應用科學科(一)

組 別：國中組



前言-研究動機

- COVID-19疫情爆發以來，**學校須每天監控學生的體溫與身體狀況**

目前校內學生體溫監控流程：

1. 每日上午與中午，進行全班體溫的量測與記錄
2. 完成後將量測結果送至學務處，由行政人員進行體溫的登錄與監控

體溫監控流程有以下缺點：

1. **耗費人力**：須有專人幫忙量測體溫
2. **量測費時**：平均一位同學約**20秒~30秒**，整班量測約**15分鐘~20分鐘**
3. **體溫登錄易誤植**：行政人員須把體溫登錄於電腦表單中，但人數眾多，易誤植量測結果

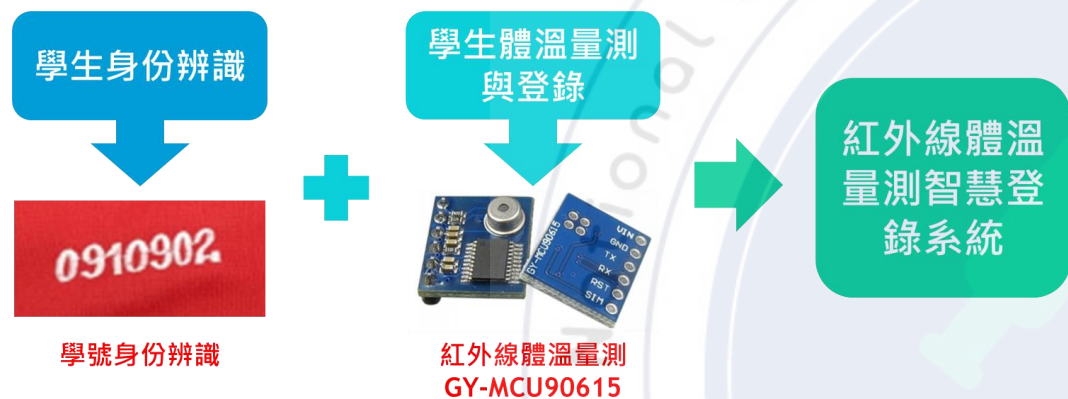
- 研究動機：透過相關研究**開發出省時省人力的體溫量測裝置**，並將其運用於校園中。

- 研究目的：**(一)學生學號身份辨識**、**(二)紅外線體溫量測**、**(三)紅外線體溫量測智慧登錄系統整合與資料處理**

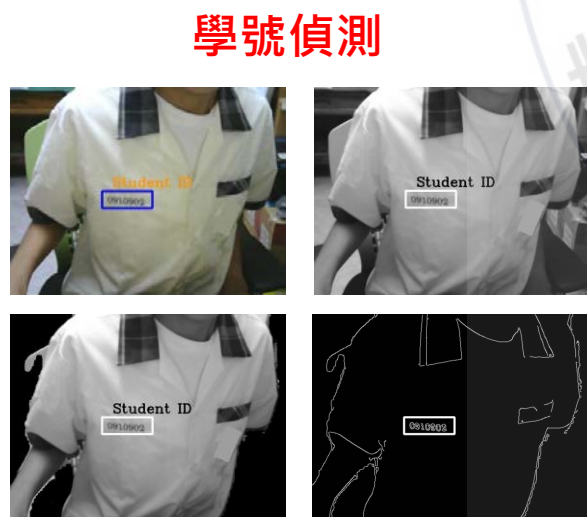


研究方法

■ 研究設計概念：



■ 學號偵測辨識優化



■ 研究工作如下：

1. Python執行OpenCV，抓取影像中的學號資訊。
2. 執行easyocr辨識學號資訊並比對學生身份。
3. 用GY-MCU90615量測體溫，並回傳至Python。
4. 將學生個資與體溫上傳網路，產出體溫監控表單。

■ 研究架構流程圖：

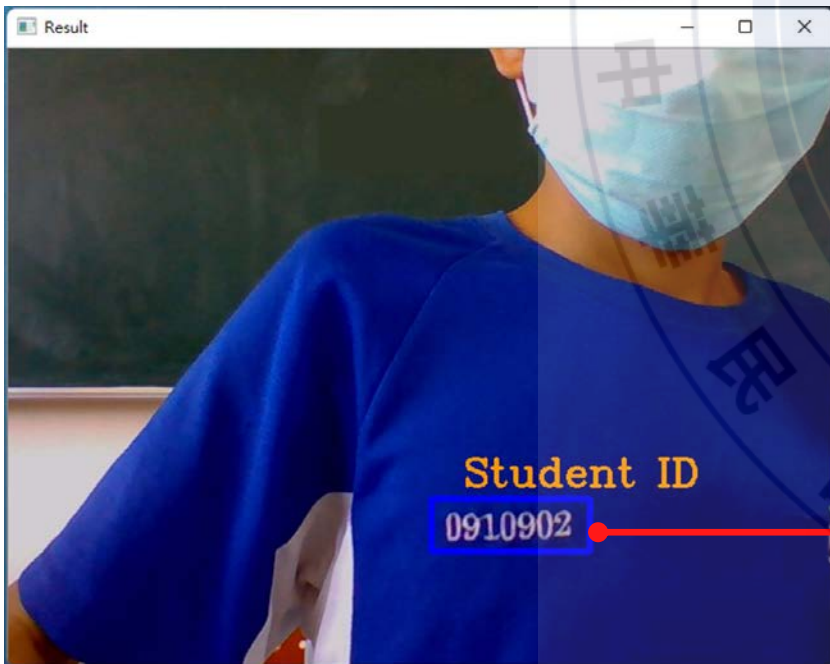


研究過程-學生學號身份辨識

✓ 實驗一、學號偵測

- 用OpenCV取得視訊鏡頭影像
- 用Harr串聯分類器偵測學號
- 框列出學號位置，擷取框列之影像

程式輸出結果(實驗一)



✓ 實驗二、學號辨識

- 將擷取學號影像用easyocr進行文字辨識



easyocr

程式輸出結果(實驗二)

```
互動環境 (Shell) ×  
Python 3.9.7 (C:\Python39\python.exe)  
>>> %Run NPD-EASYOCR.py  
  
Using CPU. Note: This module is much faster with a GPU.  
['0910902']
```

✓ 實驗三、學號身份比對

- 將辨識出的學號與學生資料比對

程式輸出結果(實驗三)

```
互動環境 (Shell) ×  
Python 3.9.7 (C:\Python39\python.exe)  
>>> %Run IR-EASYOCR-10.0-1.py  
  
Using CPU. Note: This module is much faster with a GPU.  
091090  
try again → 比對失敗  
Using CPU. Note: This module is much faster with a GPU.  
910902  
910902 209 2 何o△ → 比對成功
```

比對身份

	A	B	C	D
1	學號	姓名	班級	座號
2	910101	王o△	201	1
3	910106	馬o△	201	6
4	920122	陳o△	201	22
5	910902	何o△	209	2
6	920128	羅o△	201	28

學生資料表

研究過程-紅外線體溫量測

- ✓ 實驗四、溫度量測測試
- 使用Arduino控制GY-MCU90615進行體溫的量測，將量測結果與市售額溫槍的量測結果進行比對。

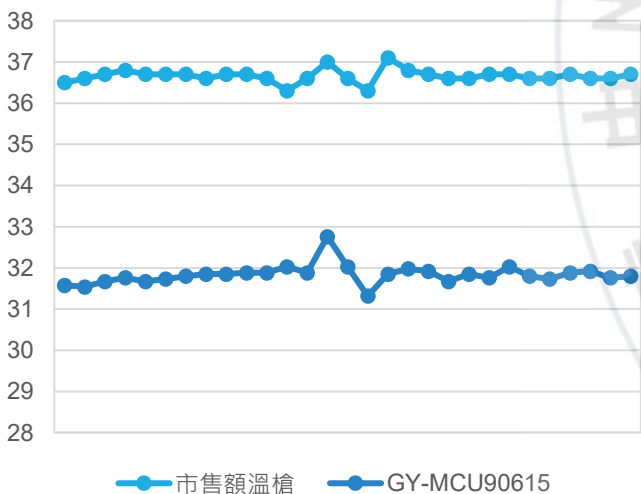
量測溫度比對實驗



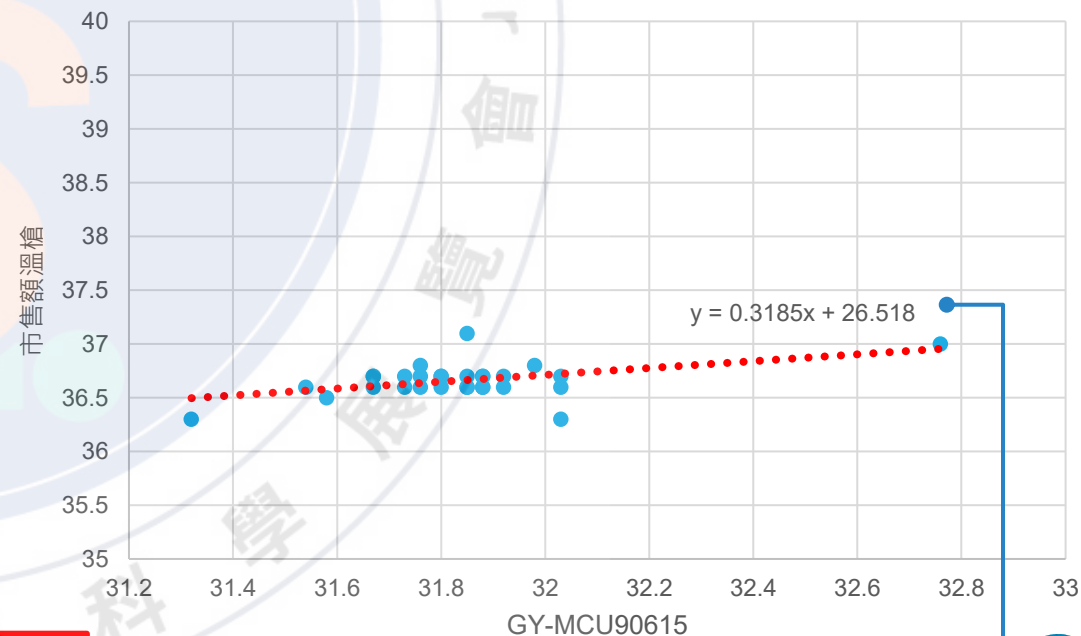
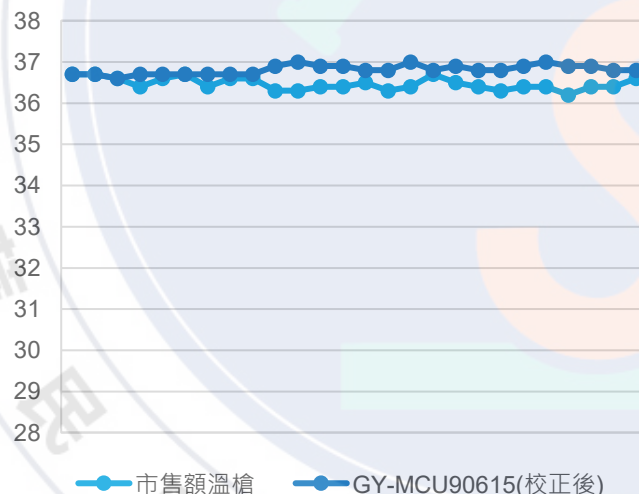
程式輸出結果

```
COM6
05:16:41.879 -> T0:26.70
05:16:47.805 -> T0:26.60
05:16:53.901 -> T0:26.79
05:17:06.767 -> T0:26.60
05:17:09.790 -> T0:26.60
05:17:15.412 -> T0:26.70
05:17:18.555 -> T0:26.75
05:17:25.124 -> T0:26.79
05:17:31.840 -> T0:26.75
05:17:34.895 -> T0:26.89
05:17:39.806 -> T0:26.89
05:17:43.891 -> T0:26.92
05:17:47.281 -> T0:28.17
05:17:50.592 -> T0:28.20
```

量測溫度比對結果(校正前)



量測溫度比對結果(校正後)



$$\text{量測額溫} = 0.3185 \times (\text{GYMCU90615量測結果}) + 26.518$$

研究過程-紅外線體溫量測

✓ 實驗五、裝置硬體設計

- 為了方便量測額溫，將GY-MCU90615用模型架架高。
- 量測裝置加裝LCD螢幕以及環狀彩燈，使量測者清楚知道自己的體溫狀況。

✓ 實驗六、裝置程式設計

- 按下量測按鈕將顯示溫度，而**37.5°C**作為體溫過高與正常的判斷標準：

未
按
按
鈕



按
下
按
鈕



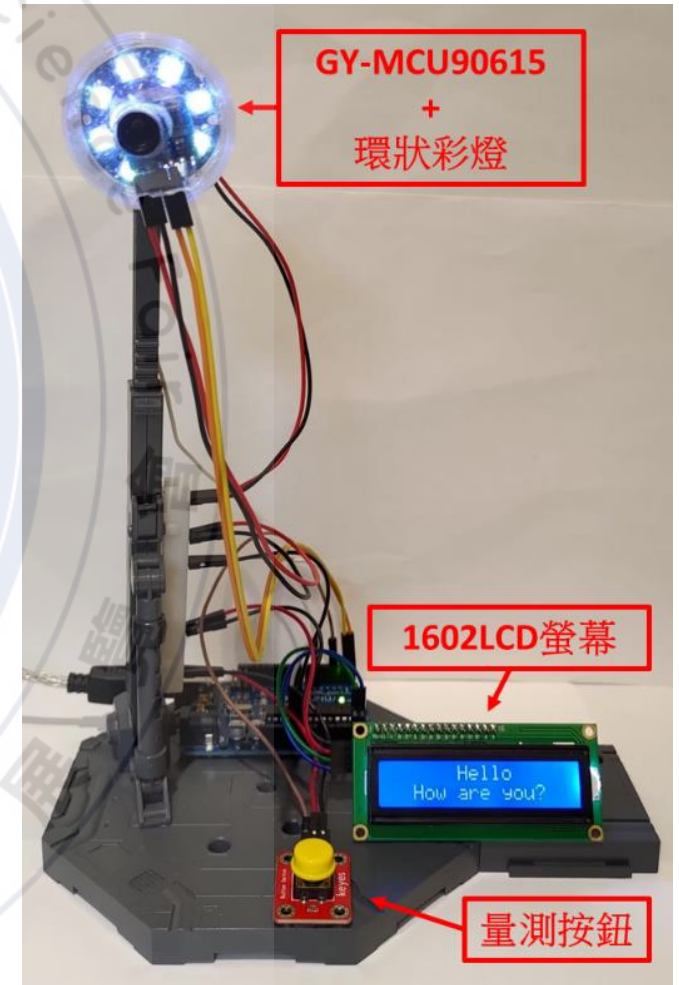
未進行體溫量測(白光)



體溫 < 37.5°C(綠光)



體溫 >=37.5°C(紅光)



研究過程-紅外線體溫量測智慧登錄系統

✓ 實驗七、資料整合

- 將GY-MCU90615所測到的**溫度**輸入python，與當下時間（包含**日期**與**時間**）以及所辨識出的學號資訊（包含**班級**、**姓名**、**座號**）結合後輸出。

程式輸出結果(實驗七)

```
互動環境 (Shell) x
35.69
日期 量測時間 學號 班級座號 姓名 溫度
2022/03/21 09:53:34 920722 207 22 廖○△ 35.69
```

✓ 實驗八、輸出表單

- 將實驗七的結果存成CSV檔，完成學生的體溫監控表單。資料登錄欄位順序設計為：**日期**、**時間**、**學號**、**班級**、**座號**、**姓名**、**溫度**。

程式輸出結果(實驗八)

	A	B	C	D	E	F	G
1	Date	Time	Student ID	Classroom	Seat number	Name	Student Temperature
2	日期	時間	學號	班級	座號	姓名	溫度
3	2022/3/18	12:28:51	920420	204	20	吳○△	37.80
4	2022/3/18	12:32:08	920125	201	25	賴○△	37.80
5	2022/3/18	12:32:41	920122	201	22	陳○△	38.07
6	2022/3/18	12:34:29	920128	201	28	羅○△	37.97
7	2022/3/18	12:35:14	910106	201	6	馬○△	38.36
8	2022/3/18	12:35:47	920217	202	17	夏○△	37.88
9	2022/3/18	12:36:07	920821	208	21	洪○△	38.10
10	2022/3/18	12:40:16	910503	205	3	王○△	37.85

✓ 實驗九、上傳網路平台

- 使用google 相關Python 套件搭配 google sheets API，將體溫監控表單即時上傳至教師google sheets，達到遠端監控的目的。

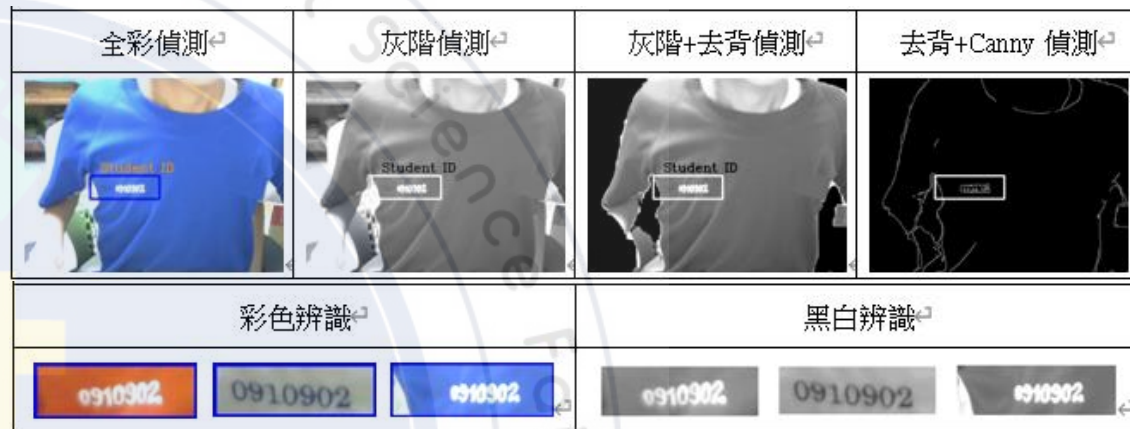
程式輸出結果(實驗九)

	A	B	C	D	E	F	G
1	日期	時間	學號	班級	座號	姓名	溫度
2	2022/03/18	12:28:51	920420	204	20	吳○△	37.8
3	2022/03/18	12:32:08	920125	201	25	賴○△	37.8
4	2022/03/18	12:32:41	920122	201	22	陳○△	38.07
5	2022/03/18	12:34:29	920128	201	28	羅○△	37.97
6	2022/03/18	12:35:14	910106	201	6	馬○△	38.36
7	2022/03/18	12:35:47	920217	202	17	夏○△	37.88
8	2022/03/18	12:36:07	920821	208	21	洪○△	38.10
9	2022/03/18	12:40:16	910503	205	3	王○△	37.85
10	2022/03/18	12:40:16	910503	205	3	王○△	37.85
11	2022/03/18	12:41:26	910909	209	9	黃○△	37.76
12	2022/03/18	12:43:45	910902	209	2	何○△	37.88
13	2022/03/18	12:44:19	920722	207	22	廖○△	37.97
14	2022/03/18	12:54:31	910705	207	5	邱○△	35.92
15	2022/03/18	13:52:54	876431	210	31	郭○△	36.29
16	2022/03/18	14:19:29	920420	204	20	吳○△	36.07
17	2022/03/18	14:19:56	920426	204	26	陳○△	36.07
18	2022/03/19	10:11:30	973232	304	38	郭○△	34.64
19	2022/03/21	09:51:48	920722	207	22	廖○△	3.29

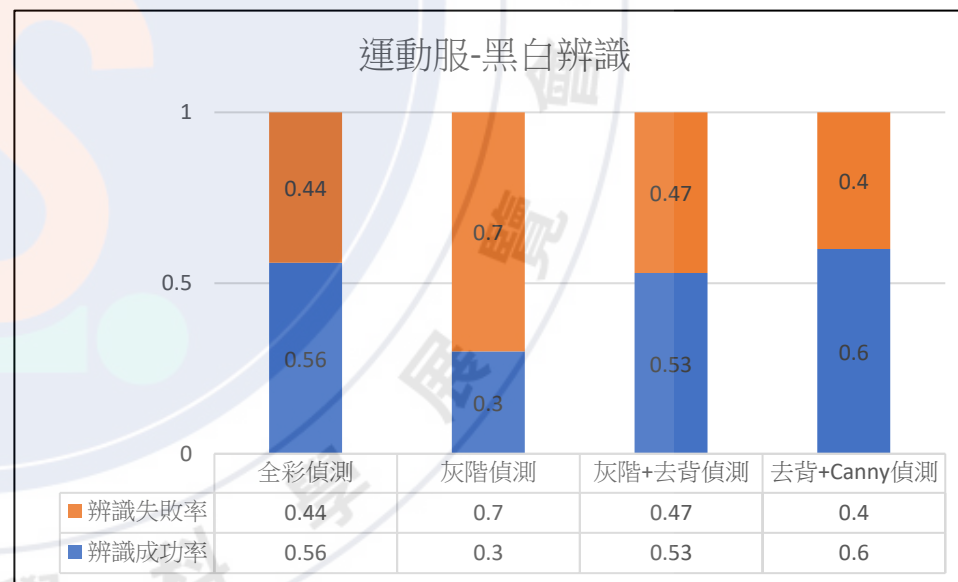
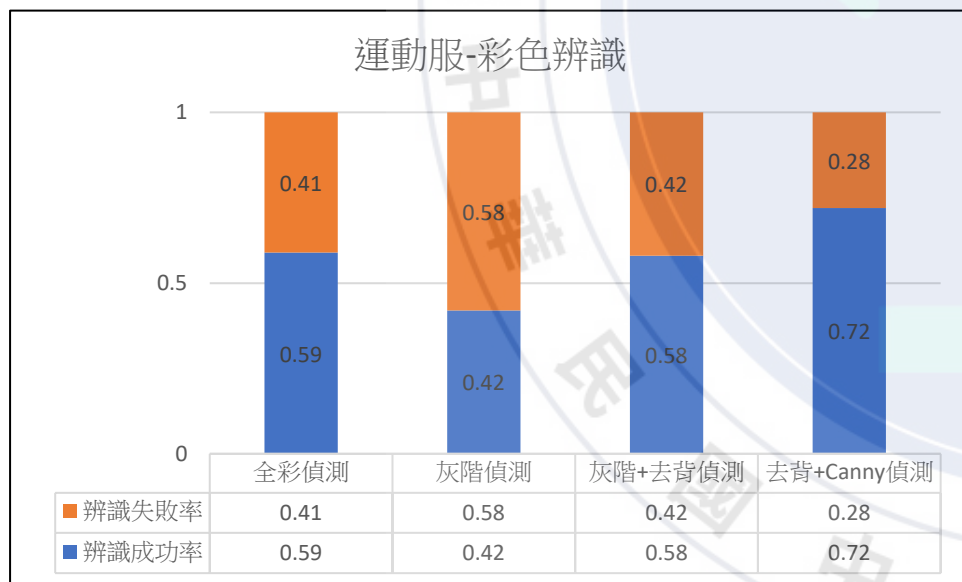
研究過程-學號偵測辨識優化

✓ 實驗十、學號偵測辨識優化

- 偵測畫面：全彩、灰階、灰階+去背、去背+Canny (邊緣檢測)
- 學號辨識：彩色辨識、黑白辨識



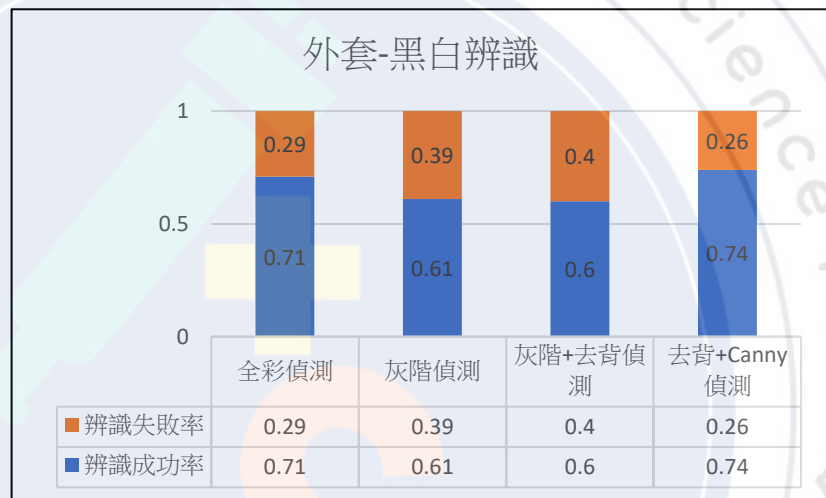
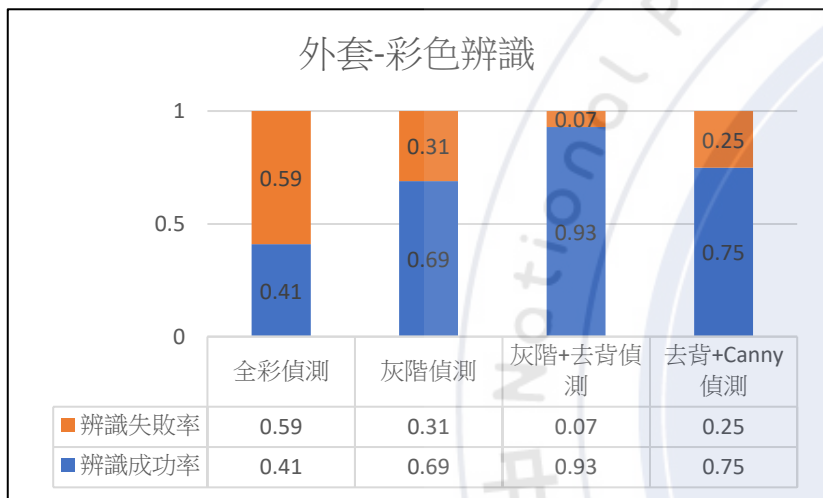
(一) 運動服偵測辨識實驗之結果



- 在這四種偵測方式下都都有類似的變化趨勢，都是(去背+Canny)偵測的成功率最高。

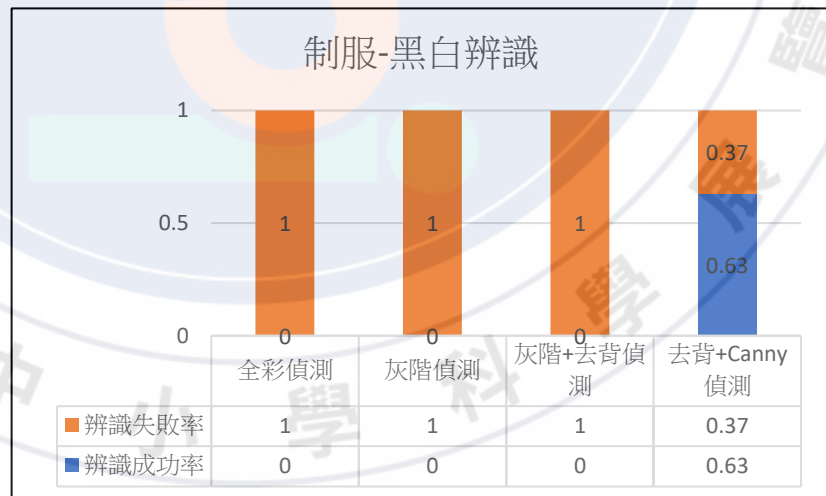
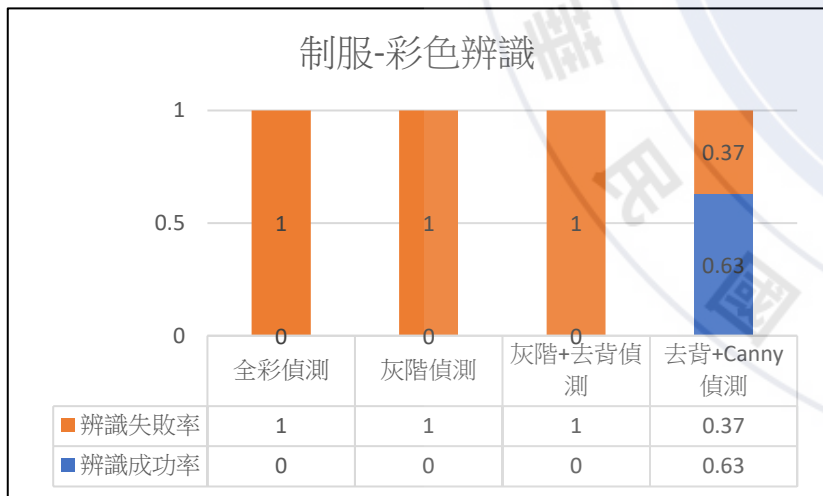
研究過程-學號偵測辨識優化

(二) 外套偵測辨識實驗之結果



外套在彩色辨識下的成功率幾乎都高於黑白辨識的成功率，且成功率幾乎都在0.6以上，其中(去背+Canny)偵測方式的成功率接近0.75。

(三) 制服偵測辨識實驗之結果



制服除了(去背+Canny)偵測的方式外，其餘的偵測方式都無法順利完成學號偵測辨識，(去背+Canny)偵測可解決制服學號無法辨識成功的問題，成功率也都高於0.6

研究結果與討論-學號偵測辨識優化

■ 根據學號偵測辨識優化實驗的結果發現：

1. (去背+Canny)偵測可以解決制服辨識的問題，在運動服跟外套上也有不錯的辨識成功率。
2. (去背+Canny)偵測 + 彩色辨識的成功率較(去背+Canny)偵測 + 黑白辨識的成功率來的更佳。

	彩色辨識	黑白辨識
運動服	0.72	0.6
外套	0.75	0.74
制服	0.63	0.63



✓ 紅外線體溫量測智慧登錄系統偵測辨識學號方法：(去背+Canny)偵測 + 彩色辨識

✓ (去背+Canny)偵測 + 彩色辨識的辨識成功平均時間可控制約在3.5秒內

	運動服	外套	制服
辨識成功率	0.72	0.75	0.63
辨識成功平均時間	3.09(s)	2.87(s)	3.15(s)

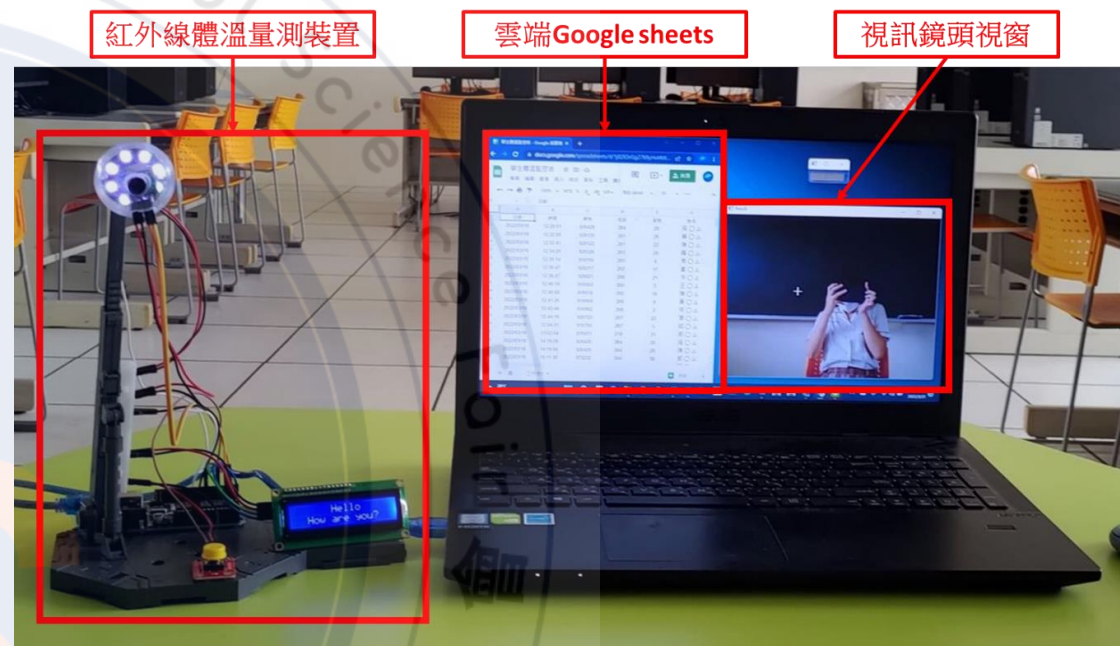
研究結果與討論-紅外線體溫量測智慧登錄系統

■ 『紅外線體溫量測智慧登錄系統』優點：

1. 可以自行操作，無須專人幫忙量測與記錄。
2. 平均一位同學量測時間約5秒，整班量測於10分鐘內完成。
3. 程式可自行輸出體溫監控表單，並上傳至網路雲端。

結論

1. 成功使用Python OpenCV套件，開發出可以自動偵測學號位置，並框列學號影像的偵測系統。
2. 成功使用Python easyocr套件進行學號影像辨識，並將影像學號轉成字串學號。
3. 成功使用Python程式將字串學號與學生身份進行比對。
4. 成功使用Arduino程式進行GY-MCU90615紅外線量測元件的測試，並與市售額溫槍進行量測結果校正。
5. 成功開發出紅外線體溫量測裝置的硬體設備。
6. 成功使用Arduino程式碼控制紅外線體溫量測裝置，並可快速進行體溫量測。



結論

7. 成功使用Python程式將學生資料與量測體溫進行整合。
8. 成功使用Python程式將整合完成的資料，輸出成體溫監控表單。
9. 成功使用Python程式與師長的google sheets API，將體溫監控表單上傳至師長的雲端google sheets，以利進行即時遠端監控。
10. 成功使用(去背+Canny)偵測+彩色辨識的方式，解決制服學號不易辨識成功的問題，使紅外線體溫量測智慧登錄系統有更佳的辨識成功率(大於0.6)與更快的辨識時間(約3.5s)。
11. 成功開發出『紅外線體溫量測智慧登錄系統』，並已運用於本校資優特色班級，未來希望能將此系統推廣至全校班級。

參考文獻

- [1] 朱克剛(2021)· AIOT與OpenCV實戰應用· 台北市· 碁峯資訊。
- [2] 郭英勝、鄭志宏、龔志銘、謝哲光 (2018)· 實用Python 程式設計· 碁峯資訊。
- [3] 楊明豐(2018)· Arduino物聯網最佳入門與應用：打造智慧家庭輕鬆學· 碁峯資訊。