

中華民國第 64 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

第三名

032803

AI 排球教練

學校名稱：臺中市立大雅國民中學

作者： 國二 吳念宇 國二 毛奐瑄 國二 張明謹	指導老師： 陳盈昌
---	------------------

關鍵詞： Python、Arduino、AI

摘要

現在流行 AI，我們就想設計一款程式來教人打排球。首先下載 python、opencv 和 mediapipe 的 Pose landmark detection guide，然後寫入程式，印出關節點的座標。接著找排球隊員，拍攝托球、低手接球、發球、扣球的影片，然後擷取碰球瞬間的影格，載入程式中並計算各個關節點之間的角度，從而定義出標準動作中的各個角度。接著拍攝初學者的各種基本動作影片，然後依照同樣的方法計算各關節點的角度，再與標準版的角度做比較，即可列印出初學者在各個關節點應該修正的角度。再利用 Arduino 及壓力傳感器做出托球矯正手套，讓人托球之後用 LED 顯示各個指頭觸球的先後順序，再讓人矯正手指的動作。藉由 AI 的操作，可以讓初學者快速知道自己的缺點，也知道修正的方法，這就是 AI 教練的魅力。

壹、前言

一、研究動機

學校在上排球體育課的時候，老師在教同學做一些基本動作，明明老師都指證了同學的動作錯在哪裡，但是同學在下次再做相同動作的同時，依然會做錯，讓老師不斷的糾正到很煩，於是我們就想到了資訊課所教的程式設計，想用影像辨識的方式來找出同學錯誤的地方以及修正的參考，期望同學對攝影機打球就會顯示自己的錯誤之處，並顯示改正的方法。

二、研究目的

- (一) 找出可以做影像辨識的程式
- (二) 找出 python 程式中與影像辨識有關的指令
- (三) 找出 opencv 與影像相關的程式碼
- (四) 用 python 與 opencv 顯示出影像
- (五) 下載 mediapipe 程式，顯示各個關節的座標點
- (六) 寫出關節彎曲角度的程式
- (七) 找出排球的各種基本動作，並找動作最標準的人拍成影片
- (八) 找出各種標準動作各關節的彎曲角度
- (九) 在初學者的影片中顯示出與標準動作的角度差
- (十) 分析初學者的錯誤地方及矯正方式
- (十一) 測量托球六個手指的碰球時間
- (十二) 製作低手傳球感測袖套
- (十三) 利用前測與後測比較研究的成果

三、文獻回顧

(一)、中華民國第 63 屆中小學科學展覽會 國小組生活與應用科學科 一
作品名稱:挺立-應用全身辨識輔助站姿調整即時回饋系列之研究

該作品是用機器學習的方式來判斷角度的變化，所以辨識度會有一些誤差。

(二)、中華民國第 61 屆中小學科學展覽會 國中組 生活與應用科學(一)
作品名稱:搶救生命大作戰- AI 姿態辨識在智慧型高品質 CPR 訓練引導
式

教學輔具系統設計之研究

該作品是利用各種感測器來偵測操作者的動作是否正確。

(三)、中華民國第 60 屆中小學科學展覽會 高級中等學校組 工程學(一)
科

作品名稱:基於 OpenPose 之機具操作於異常姿勢即時偵測

該作品是利用影像辨識出各個關節點的座標，在利用高中的數學來計算角度，從而判斷姿勢是否正確。

貳、 研究設備及器材

一、Arduino 如圖一

圖一: Arduino(第一作者
拍攝)



二、壓力感測器 如圖二

圖二:壓力感測器(第一作
者拍攝)



三、AI 智慧手套正、反面，如圖三、四

圖三:AI 智慧手套正面(第一作者拍攝)

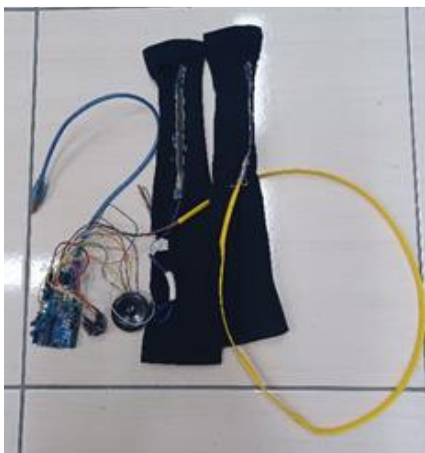


圖四:AI 智慧手套反面(第一作者拍攝)



四、AI 智慧袖套正面，如圖五

圖五:AI 智慧袖套正面(第一作者拍攝)



參、 研究過程和方法

一、找出可以做影像辨識的程式

1. 上網搜尋跟影像辨識有關的程式
2. 篩選適合程度的軟體並下載
3. 實際測試可行的程度
4. 選擇適用的軟體

二、找出 python 程式中與影像辨識有關的指令

1. 下載 python 程式
2. 找出計算距離和角度的相關程式碼
3. 練習計算點與點之間的距離以及兩條線的角度

三、找出 opencv 與影像相關的程式碼

1. 下載 opencv 程式
2. 建立一個空畫面
3. 找出與繪圖有關的程式碼並執行
4. 找出在圖片寫文字的程式碼並執行

四、用 python 與 opencv 顯示出影像

1. 在 opencv 程式中載入一個影像檔
2. 在影像上面繪圖與寫字

五、下載 mediapipe 程式，顯示各個關節的座標點

1. 到 mediapipe 網站下載有關於全身座標的程式
2. 適當修改程式碼印出特定座標的程式

六、寫出關節彎曲角度的程式

1. 找出低手接球相關座標點
2. 利用三角函數計算關節彎曲的角度

七、找出排球的各種基本動作，並找動作最標準的人拍成影片

1. 詢問排球教練有哪些基本動作
2. 請排球教練挑一個動作最標準的選手來拍影片

八、找出各種標準動作的關節彎曲角度

1. 從拍攝的影片中擷取碰球瞬間的照片
2. 將截圖送至 python 檔案並計算各關節的彎曲角度

九、在初學者的影片中顯示出與標準動作的角度差

1. 找一位初學者來拍攝各種排球動作的影片
2. 擷取碰球瞬間的照片
3. 將截圖送至 python 檔案並計算各關節的彎曲角度

十、分析初學者的錯誤地方及矯正方式

1. 比較初學者與標準版的關節彎曲角度差
2. 告知初學者需要修正的角度
3. 請初學者練習幾次後再拍成影片直到完全校正為止

十一、測量托球六個手指的碰球時間

1. 在手套上縫製壓力感測器及 LED 燈泡
2. 連結至 arduino。
3. 寫程式設定當觸球的時候，紀錄那根手指的觸球時間。
4. 觸球後，延遲 3 秒，讓燈光發亮 6 秒。
5. 利用燈泡亮起的時間比較觸球時間是否相同

十二、製作低手傳球感測袖套

1. 把壓力感測器一端連接 5v、一端連接電阻及 GND，然後再連接到 A0 作為右手的感測器。左手則連接到 A1，作為左手的感測器。
2. 將壓力感測器用塑膠套包起來後，縫在袖套上，縫製的地方要對準

擊球點。

3. 將 mp3 播放模組的接線連接到 Arduino
4. 寫入程式碼，當兩個擊球點碰撞的時間在 0.01 秒以內就發布正確的訊號。如果右手的時間比左手的時間晚了 0.01 秒以上，就發布右手高一點的訊號，反之則發布右手低一點的訊號。

十三、利用前測與後測比較研究的成果

1. 前測
2. 後測
3. 比較

肆、研究結果

一、找出可以做影像辨識的程式

(一) 上網搜尋跟影像辨識有關的程式

1. 機器學習範例程式說明

此方法主要是利用大量的圖片讓機器學習從而做出正確的影像判斷，但是對於同樣的姿勢只有細微差別的動作卻是無法分辨，所以不適合本實驗的需求。

2. Steam 教育學習網

此方法是用 python 軟體，載入 opencv、mediapipe 程式，可以列印出關節點的座標，然後自己再利用數學公式計算出關節點的角度，很適合本實驗的需求，故採用本方法

(二) 篩選適合程度的軟體並下載

1. 下載 python 程式

(1) 到 python 官網，選擇 [Windows installer \(64-bit\)](#) 下載到電腦裡面。

(2) 開啟下載的程式並安裝，安裝過程中要勾選 path 選項。

2. 下載 visual studio code 程式，到官網

選擇 Download for windows，下載完成後安裝

3. 下載 opencv 程式

打開 visual studio code 程式，在終端機畫面鍵入 `pip install opencv-python` 即完成

4. 下載 mediapipe 程式

打開 visual studio code 程式，在終端機畫面鍵入 `pip install mediapipe` 即完成。接著到 mediapipe 的網站，下載姿勢的程式碼如附錄所示。

二、找出 python 程式中與影像辨識有關的指令

(一) 找出計算距離和角度的相關程式碼

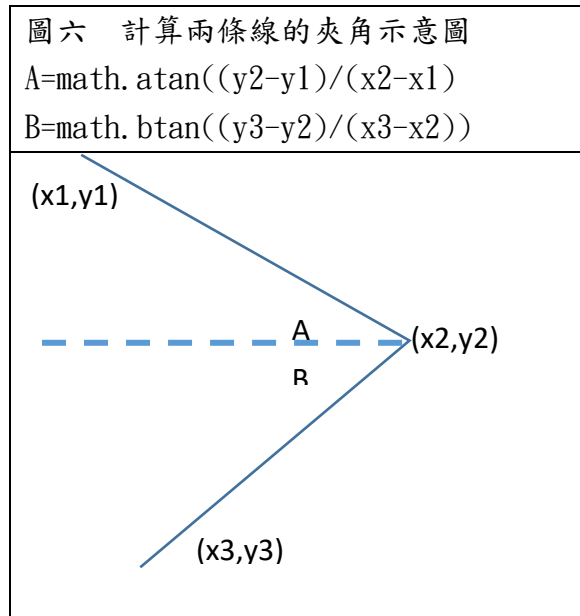
1. 下載 math 程式庫
2. 選擇常用的數學函數如下：
 - (1) a^b 的表示法: `math.pow(a, b)`。
 - (2) `sina=math.sin(a)`
 - (3) `cosa=math.cos(a)`
 - (4) `tana=math.tan(a)`

(二) 練習計算點與點之間的距離

1. 兩點的座標設為 $(x1, y1)$, $(x2, y2)$
2. 這兩點的距離數學方程式為 $\sqrt{(x1-x2)^2 + (y1-y2)^2}$
3. 把式子寫成 python 程式為
 - $(x1-x2)^2 = \text{math.pow}((x1-x2), 2)$
 - $(y1-y2)^2 = \text{math.pow}((y1-y2), 2)$
 - 兩點的距離 $= \text{math.pow}((x+y), 0.5)$

(三) 計算兩條線的角度

1. 假設兩條線的三個點的座標分別為 $(x1, y1)$, $(x2, y2)$, $(x3, y3)$ 如圖所示



三、找出 opencv 與影像相關的程式碼

- (一) 下載 opencv 程式
- (二) 建立一個空畫面
- (三) 找出與繪圖有關的程式碼並執行
- (四) 找出在圖片寫文字的程式碼並執行

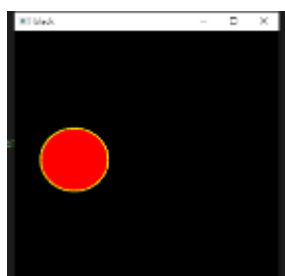
```
import numpy as np
import cv2
# 開一個空畫面:高 600, 寬 300
img = np.zeros((600, 300, 3), np.uint8)
#cv2.circle(影像, 圓心座標, 半徑, 顏色, 線條寬度)
cv2.circle(img, (90, 210), 50, (0, 255, 255), 3)
text = 'Hello, OpenCV'
# 使用各種字體 cv2.putText(影像, 文字, 第一字左下角坐標,
# 字形, 大小, 顏色, 寬度)
cv2.putText(img, text, (10, 100), cv2.FONT_HERSHEY_SIMPLEX,
2, (0, 255, 255), 5, cv2.LINE_AA)
```

四、用 python 與 opencv 顯示出影像

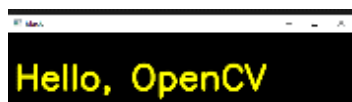
- (一) 在 opencv 程式中載入一個 g1 影像檔並在影像上面寫字, 程式碼如下

```
img1=cv2.imread('g1.png')
text = '(161, 200, 249)'
cv2.putText(img1, text, (10, 100), cv2.FONT_HERSHEY_SIMPLEX,
2, (0, 255, 0), 5, cv2.LINE_AA)
```

圖七：畫圓
(第一作者繪製)



圖八：顯示文字示意圖
(第一作者製作)



圖九：正面接球(第一作者拍攝)



五、下載 mediapipe 程式，顯示各個關節的座標點

(一)到 mediapipe 網站下載有關於全身座標的程式

```
import cv2
import mediapipe as mp
mp_drawing = mp.solutions.drawing_utils          # mediapipe 繪圖方法
mp_drawing_styles = mp.solutions.drawing_styles # mediapipe 繪圖樣式
mp_holistic = mp.solutions.holistic              # mediapipe 全身偵測方法
cap = cv2.VideoCapture(0)
# mediapipe 啟用偵測全身
with mp_holistic.Holistic(
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as holistic:
    if not cap.isOpened():
        print("Cannot open camera")
        exit()
    while True:
        ret, img = cap.read()
        if not ret:
            print("Cannot receive frame")
            break
        img = cv2.resize(img, (520, 300))
        img2 = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # 將 BGR 轉換成 RGB
        results = holistic.process(img2)           # 開始偵測全身
        # 面部偵測，繪製臉部網格
        mp_drawing.draw_landmarks(
            img,
            results.face_landmarks,
            mp_holistic.FACEMESH_CONTOURS,
            landmark_drawing_spec=None,
            connection_drawing_spec=mp_drawing_styles
                .get_default_face_mesh_contours_style())
        # 身體偵測，繪製身體骨架
```

```

mp_drawing.draw_landmarks(
    img,
    results.pose_landmarks,
    mp_holistic.POSE_CONNECTIONS,
    landmark_drawing_spec=mp_drawing_styles
        .get_default_pose_landmarks_style())
cv2.imshow('oxxostudio', img)
if cv2.waitKey(5) == ord('q'):
    break    # 按下 q 鍵停止
cap.release()
cv2.destroyAllWindows()

```

(二)適當修改程式碼印出特定座標的程式

1. 修改後程式碼如附錄。

六、寫出關節彎曲角度的程式

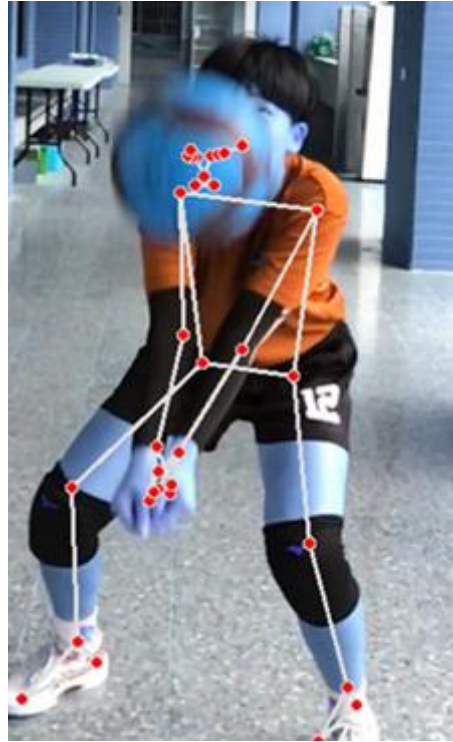
(一)找出低手接球相關座標點

<p>圖十：接球側面(第三作者拍攝)</p>	<p>圖十一：接球側面關節點(第三作者拍攝)</p>
	

圖十二：接球正面(第三作者拍攝)

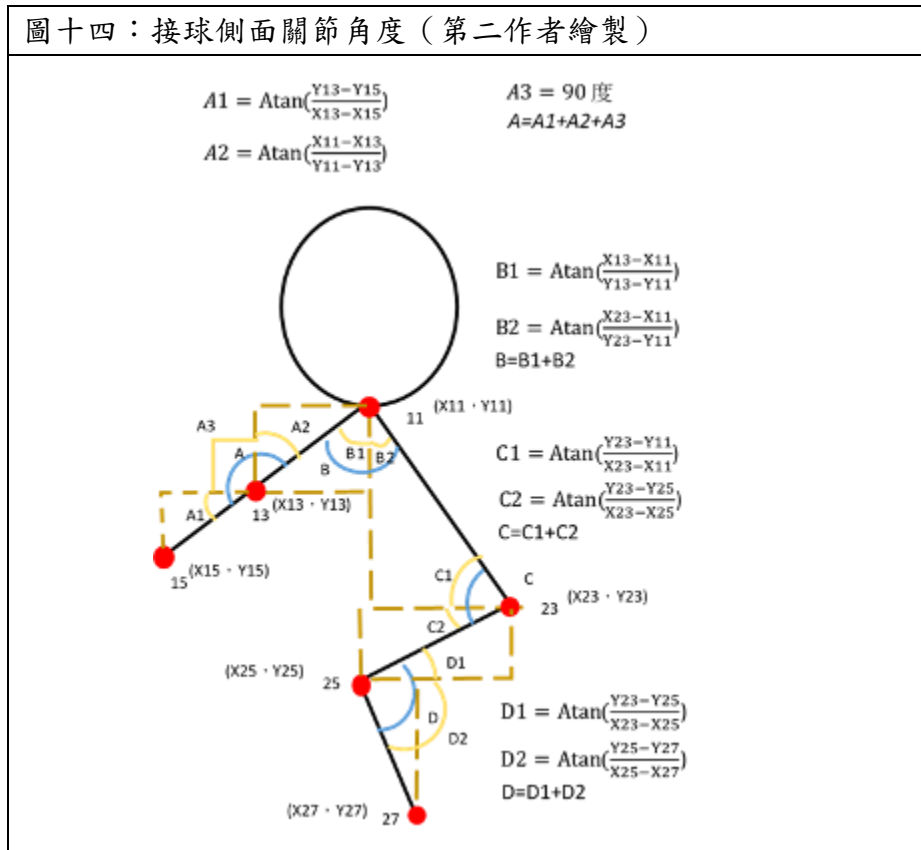


圖十三：接球正面關節點(第三作者拍攝)



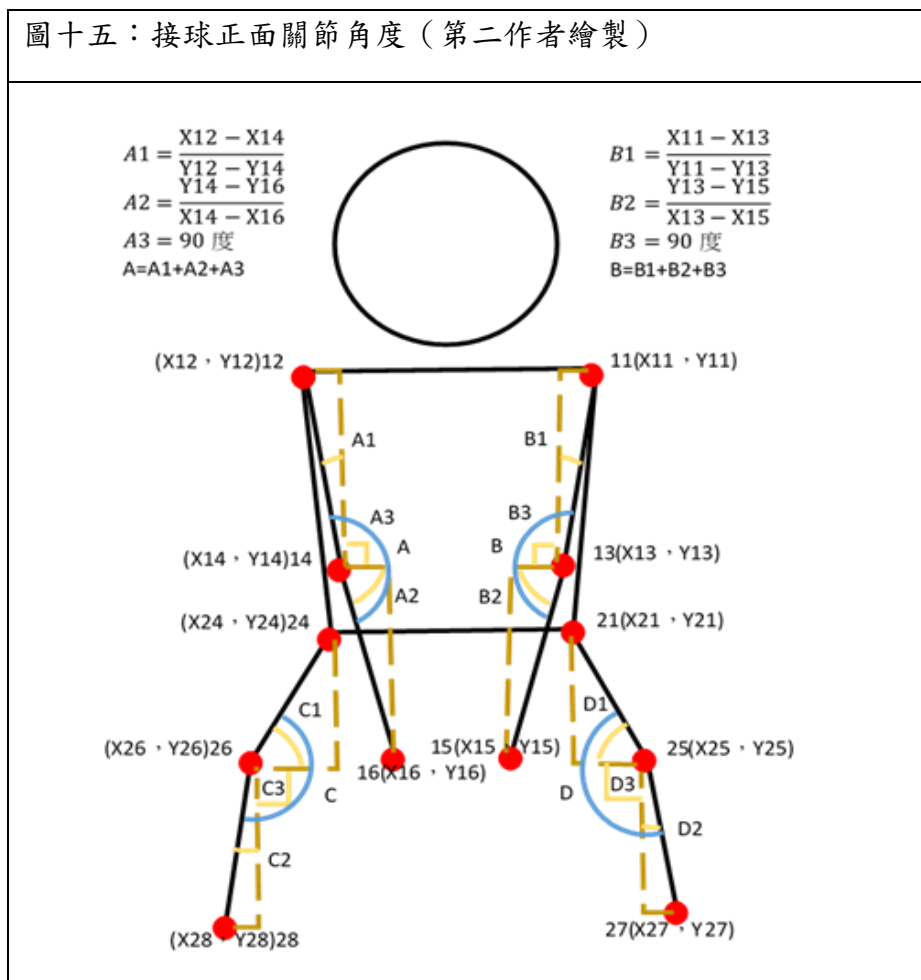
(二)利用三角函數計算關節彎曲的角度

1. 接球正面角度計算



2. 接球側面角度

圖十五：接球正面關節角度（第二作者繪製）



(三)接球程式碼

1. 接球側面(詳見附錄一)
2. 接球正面(詳見附錄二)

七、找出排球的各種基本動作，並找動作最標準的人拍成影片

(一)拍攝排球基本動作

圖十六：
發球側面(第三作者拍攝)



圖十八：
托球正面(第三作者拍攝)

圖十七：
攻擊正面(第三作者拍攝)



圖十九：
托球側面(第三作者拍攝)



八、找出各種標準動作的關節彎曲角度

(一)發球角度計算

<p>圖二十： 發球關節點 (第三作者拍攝)</p>	<p>圖二十一： 發球關節角度 (第二作者繪製)</p>
	<p> $A1 = \frac{Y15 - Y13}{X15 - X13}$ $A2 = \frac{Y13 - Y11}{X13 - X11}$ $A = A1 + A2$ </p> <p> $B1 = \frac{Y13 - Y11}{X13 - X11}$ $B2 = \frac{Y11 - Y23}{X11 - X23}$ $B3 = 90^\circ$ $B = B1 + B2 + B3$ </p>

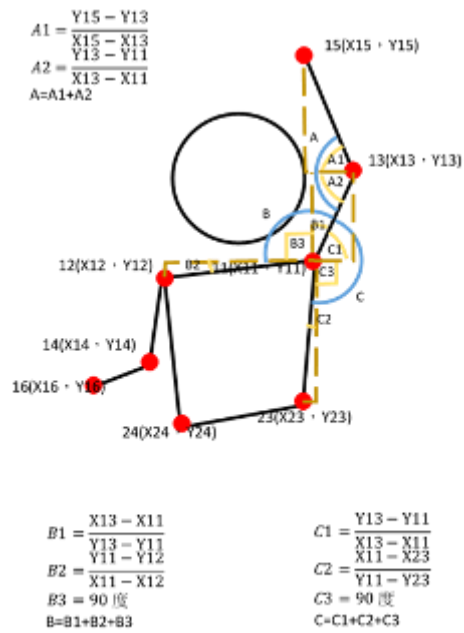
(二)發球側面程式碼(詳見附錄三)

(三) 攻擊角度計算

圖二十二：攻擊關節點
(第三作者拍攝)



圖二十三：
攻擊關節角度 (第二作者繪製)



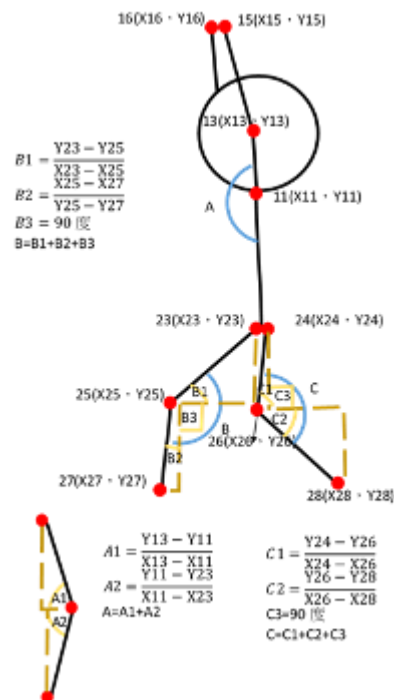
(四) 攻擊正面程式碼(詳見附錄)

(五) 托球側面角度計算

圖二十四：
托球側面關節點
(第三作者拍攝)

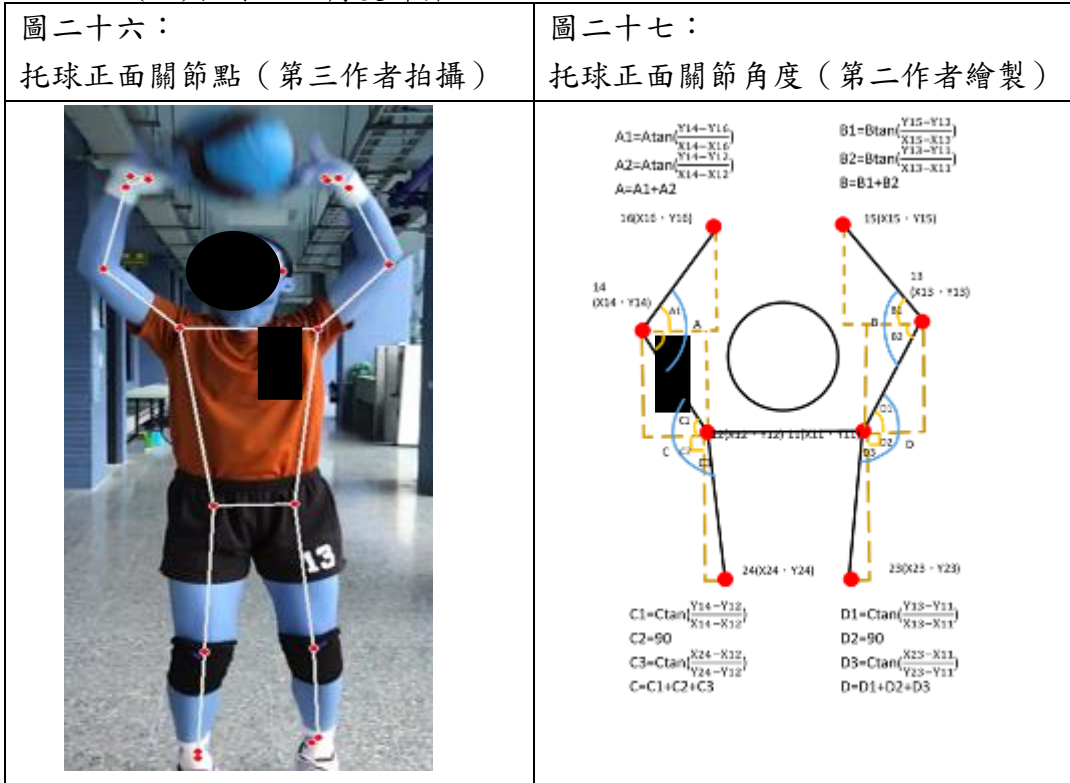


圖二十五：
托球側面關節角度 (第二作者繪製)



(六) 托球側面程式碼(詳見附錄五)

(七)托球正面角度計算



(八)托球側面程式碼(詳見附錄)

九、在初學者的影片中顯示出與標準動作的角度差

- (一)找一位初學者來拍攝各種排球動作的影片
- (二)擷取碰球瞬間的照片



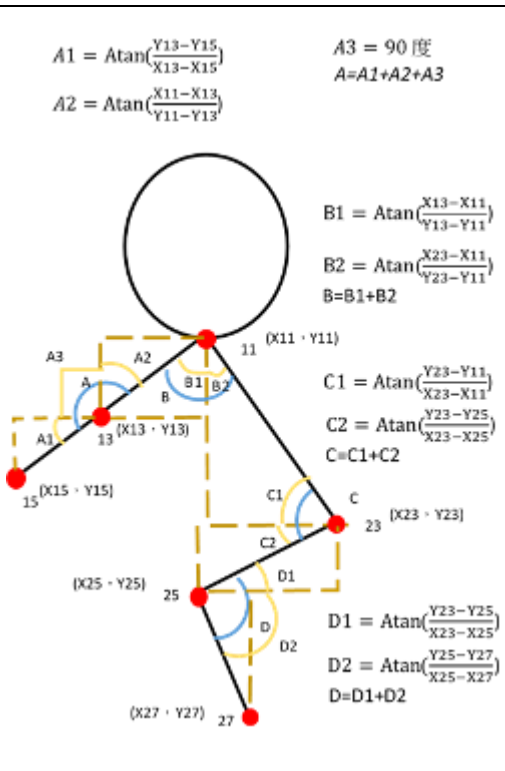
圖三十：
初學者托球側面（第一作者拍攝）



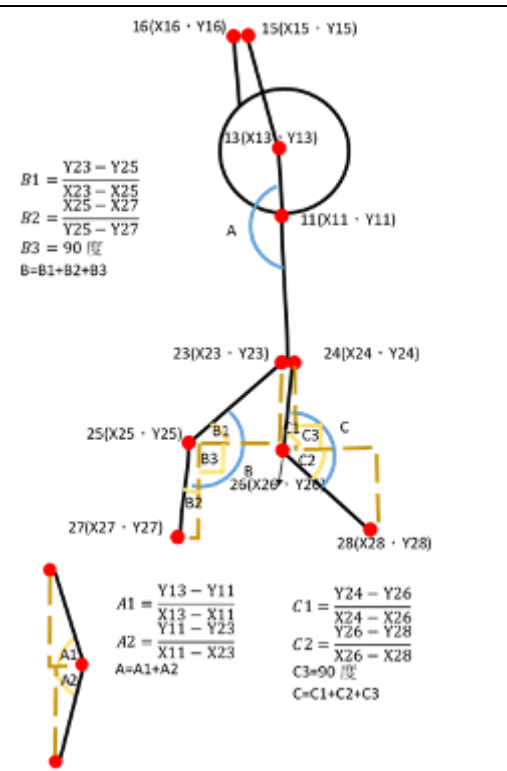
圖三十一：
初學者托球側面關節點（第一作者拍攝）



圖三十二：
接球側面關節角度（第二作者繪製）



圖三十三：
托球側面關節角度（第二作者繪製）



十、分析初學者的錯誤地方及矯正方式

- (一)比較初學者與標準版的關節彎曲角度差
- (二)告知初學者需要修正的角度
- (三)請初學者練習幾次後再拍成影片直到完全校正為止

十一、測量托球六個手指的碰球時間

- (一)在手套上縫製壓力感測器及 LED 燈泡，如圖三十四、三十五

圖三十四：
AI 智慧手套正面(第一作者拍攝)

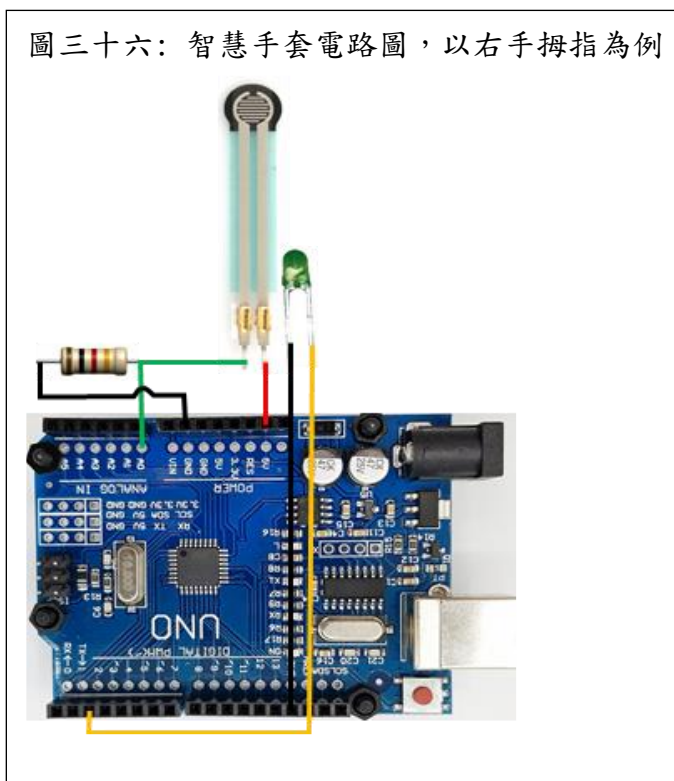


圖三十五：
AI 智慧手套反面(第一作者拍攝)



- (二)電路圖，如圖三十六

圖三十六：智慧手套電路圖，以右手拇指為例

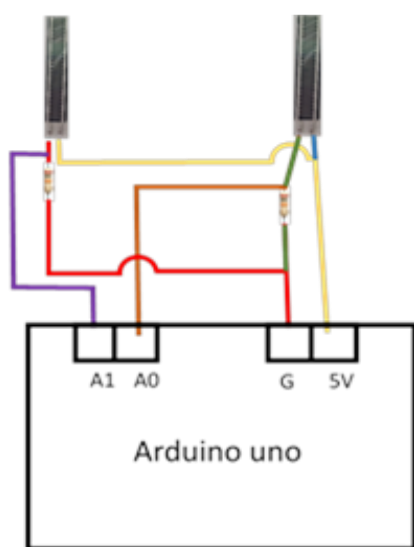


- (三)程式碼(詳見附錄七)

十二、製作低手傳球感測袖套

- (一)把壓力感測器一端連接 5v、一端連接電阻及 GND，然後再連接到 A0 作為右手的感測器。左手則連接到 A1，作為左手的感測器。如圖三十七、三十八

圖三十七:低手壓力感測器線路圖
(第二作者繪製)



圖三十八:低手傳球感測袖套展示圖
(第一作者拍攝)



- (二)將壓力感測器用塑膠套包起來後，縫在袖套上，縫製的地方要對準擊球點
- (三)將 mp3 播放模組的接線連接到 Arduino
- (四)寫入程式碼，當兩個擊球點碰撞的時間在 0.01 秒以內就發布正確的訊號。如果右手的時間比左手的時間晚了 0.01 秒以上，就發布右手高一點的訊號，反之則發布右手低一點的訊號

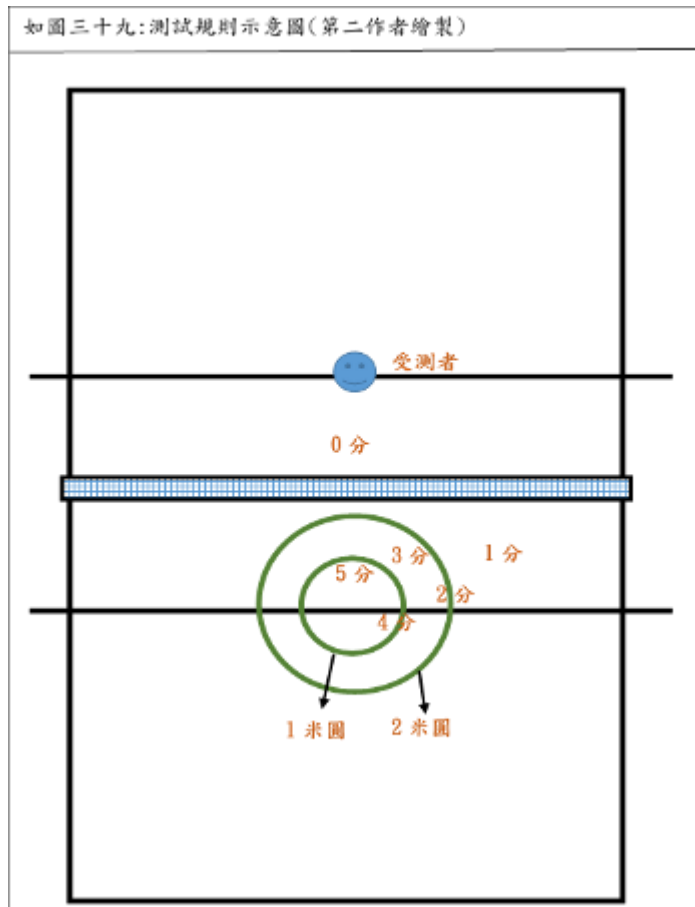
十三、利用前側與後測比較研究的成果

- (一)前測:

1. 找出每班同學三男三女，其排球程度分別是好的、中的跟比較差的
2. 每位同學打十球，再根據球的落點計算出總分，如表一、圖三十九

表一	
分數	位置
0分	未過網
1分	接球者在過網界外或未進圈
2分	接球者跨越2米圓
3分	接球者在1米圓到2米圓之間
4分	接球者跨越1米線
5分	接球者在1米圓內或一腳踩在圓上

如圖三十九：測試規則示意圖(第二作者繪製)



前測					
受測者(男)	A	B	C	D	E
托球	15分	38分	50分	38分	46分
低手	8分	17分	28分	27分	36分

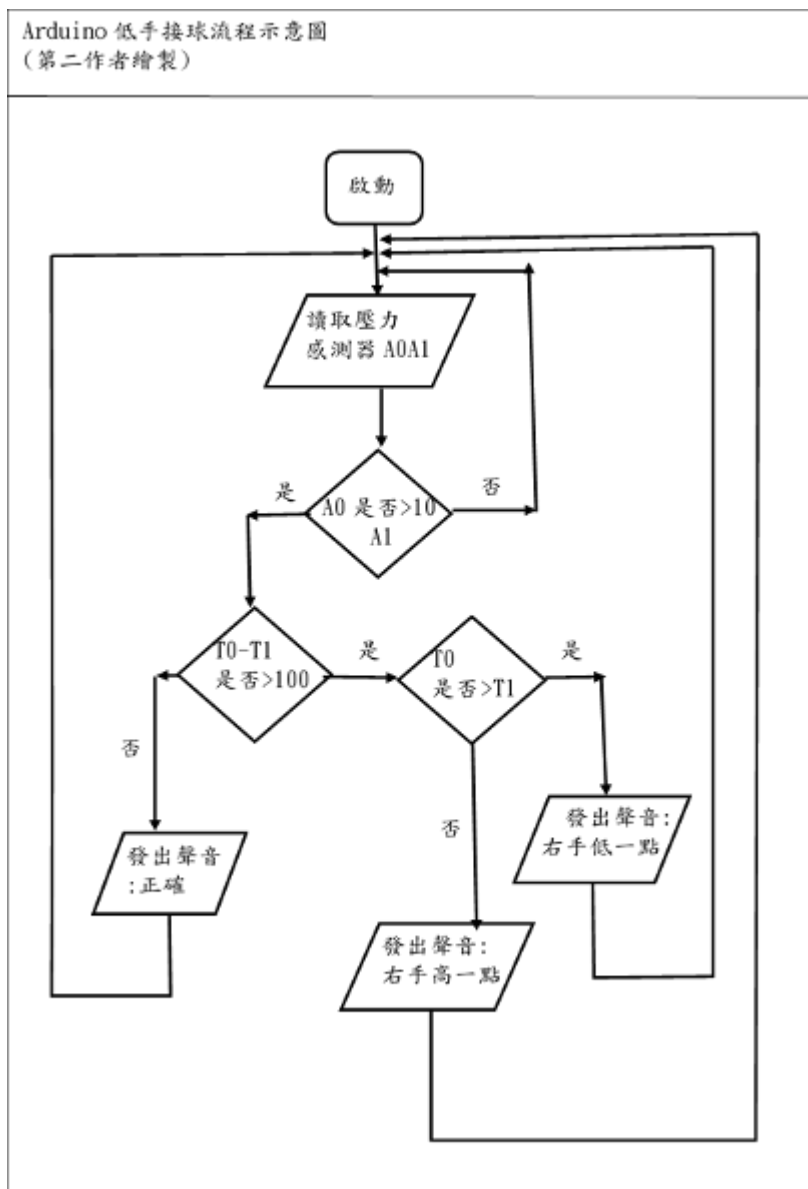
前測					
受測者(女)	A	B	C	D	E
托球	9分	34分	10分	7分	17分
低手	3分	12分	13分	3分	19分

(二)後測:

1. 低手接球

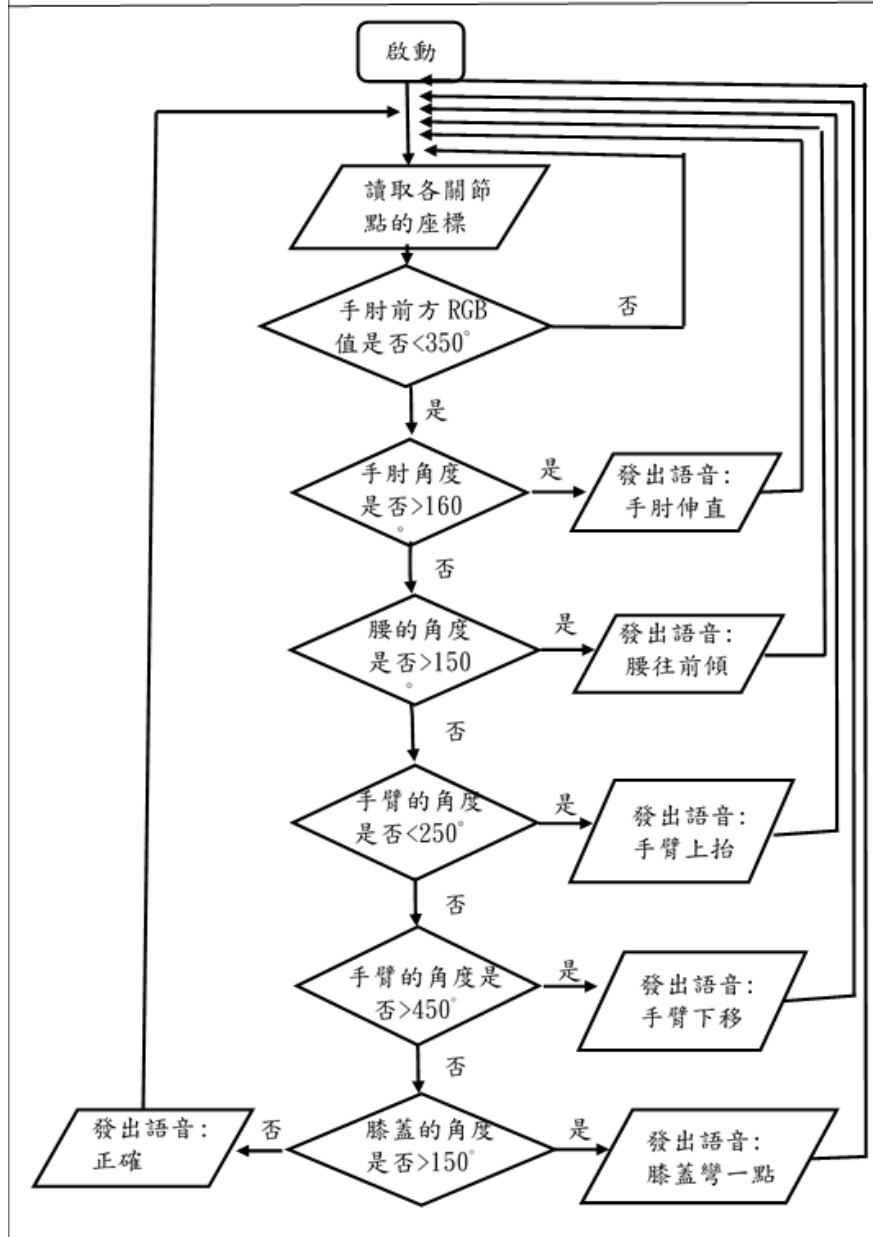
(1) 畫出 AI 教練流程圖

ㄣ. python 流程圖



ㄣ. Arduino 流程圖

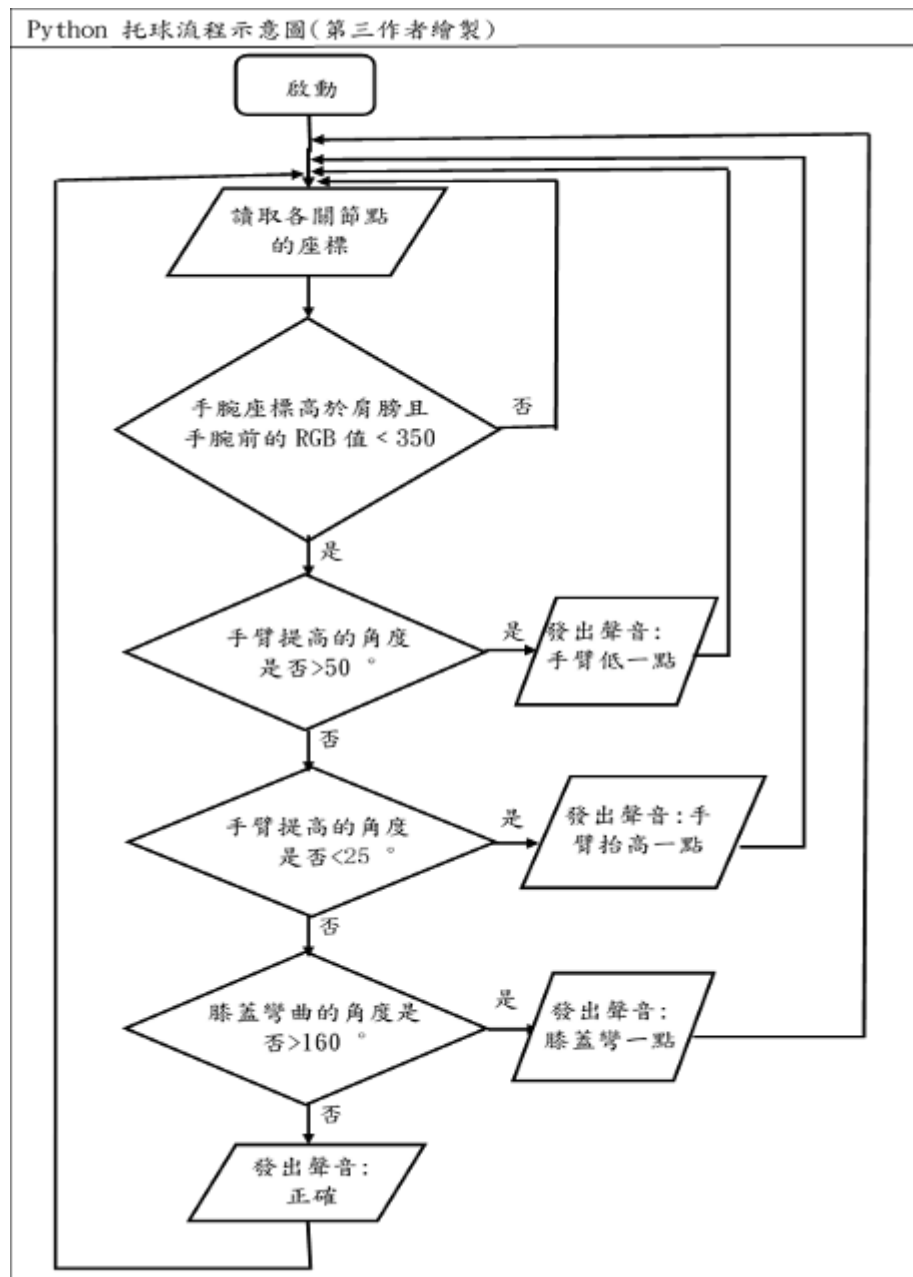
Python 低手接球流程示意圖
(第二作者繪製)



2. 托球

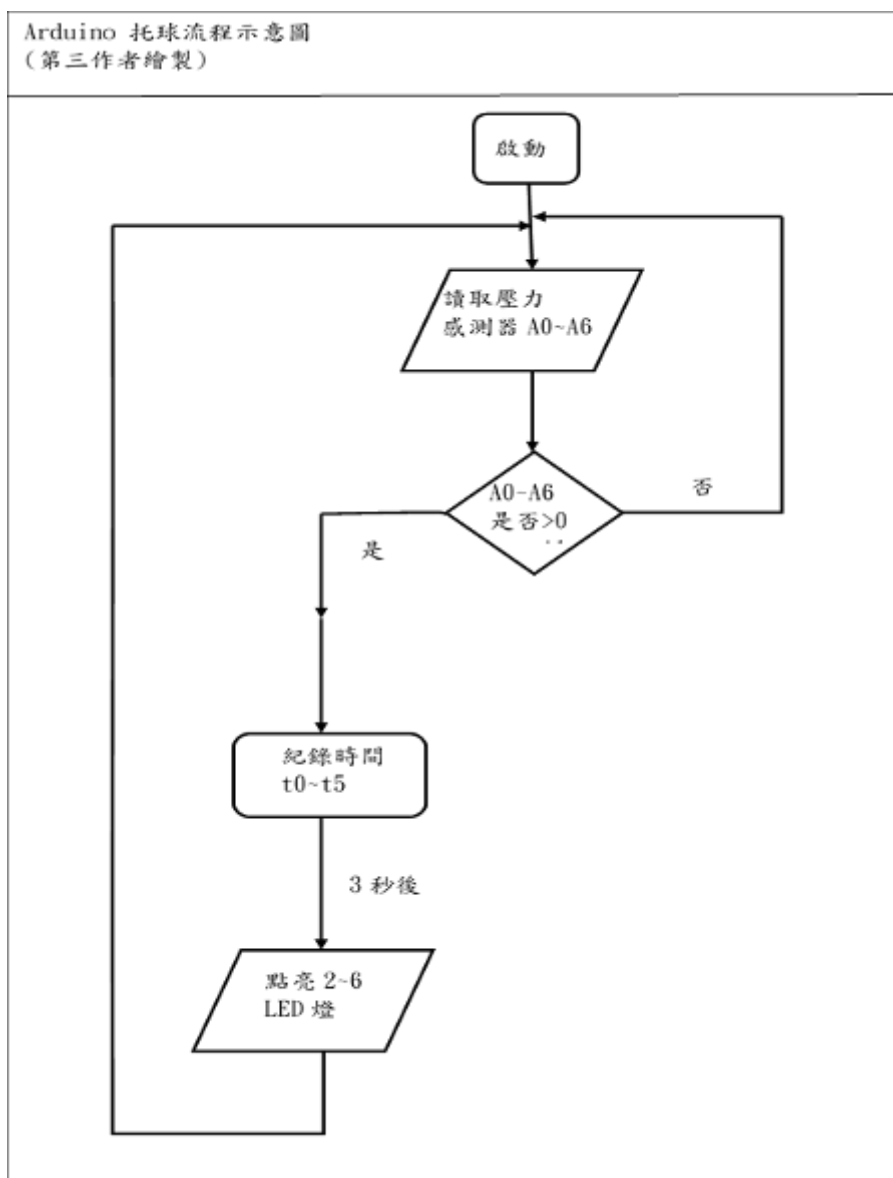
(1) 畫出 AI 教練流程圖

ㄎ. python 流程圖



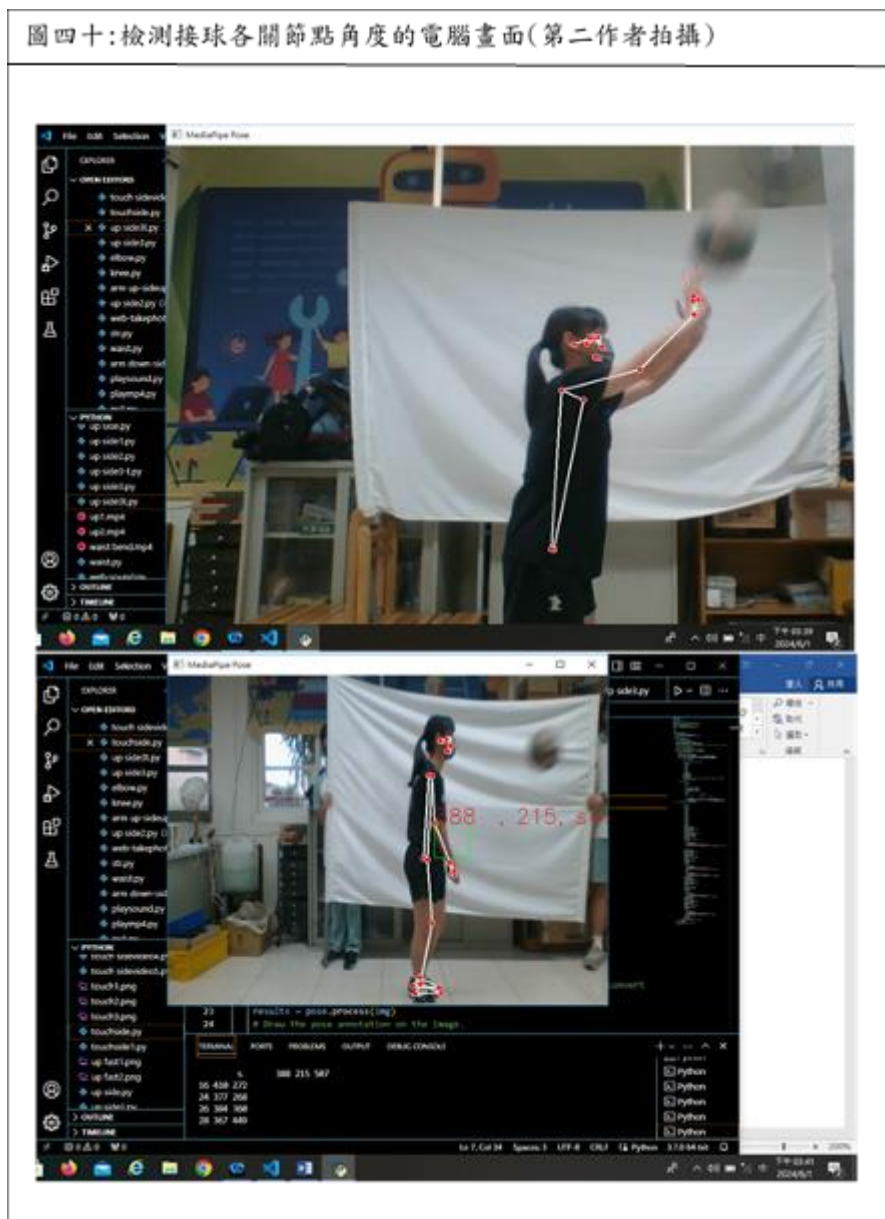
4. Arduino 流程圖

Arduino 托球流程示意圖
(第三作者繪製)



3. 找出前測的同學做後測
4. 請同學站在白布前面，讓電腦的攝像，如圖四十

圖四十：檢測接球各關節點角度的電腦畫面(第二作者拍攝)



5. 開始進行練習，拋球後電腦會根據接球者的姿勢發出指示的聲音，提醒受測者做改進
6. 練習二十球後開始進行後測

python 後測

受測者(男)	A	B	C	D	E
托球	20分	42分	50分	43分	49分
低手	9分	23分	35分	34分	38分

python 後測					
受測者(女)	A	B	C	D	E
托球	15 分	41 分	15 分	33 分	45 分
低手	9 分	24 分	27 分	25 分	38 分

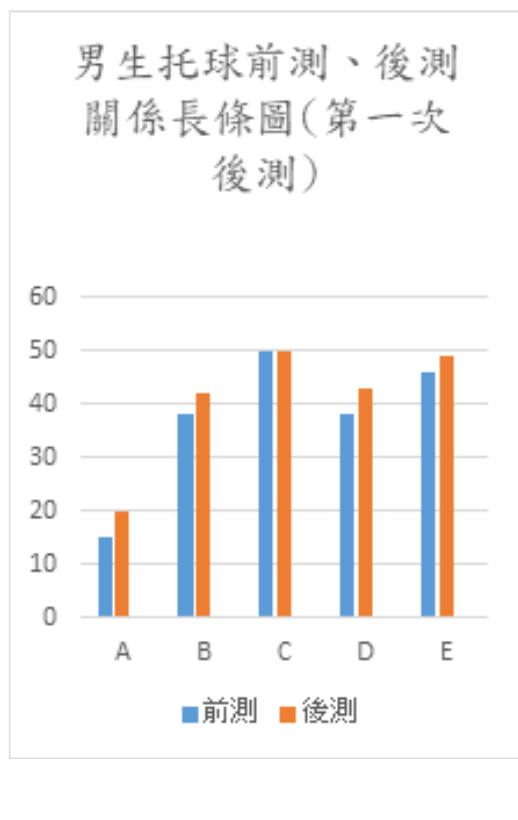
7. 戴上穿戴式裝置進行第二次後測

穿戴式裝置後測					
受測者(男)	A	B	C	D	E
托球	21 分	43 分	50 分	42 分	50 分
低手	10 分	20 分	33 分	32 分	40 分

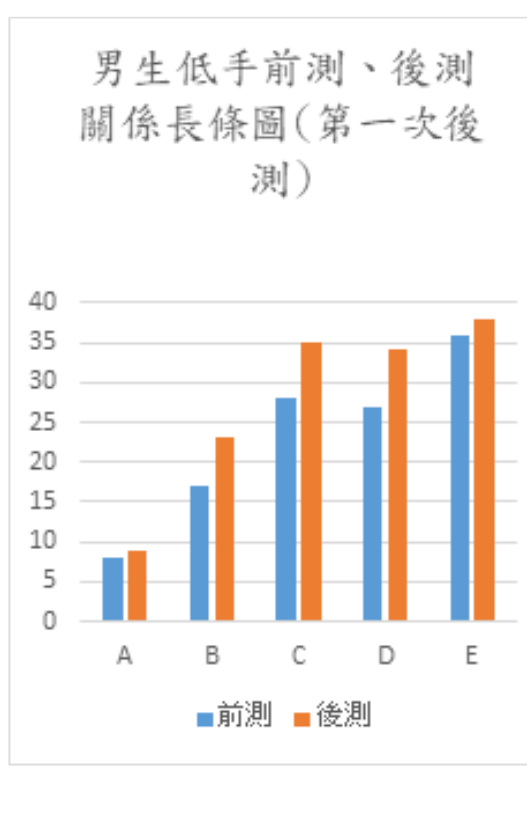
穿戴式裝置後測					
受測者(女)	A	B	C	D	E
托球	13 分	40 分	14 分	28 分	39 分
低手	9 分	22 分	25 分	20 分	36 分

8. 打完十球後統計成績，再與前測做比較

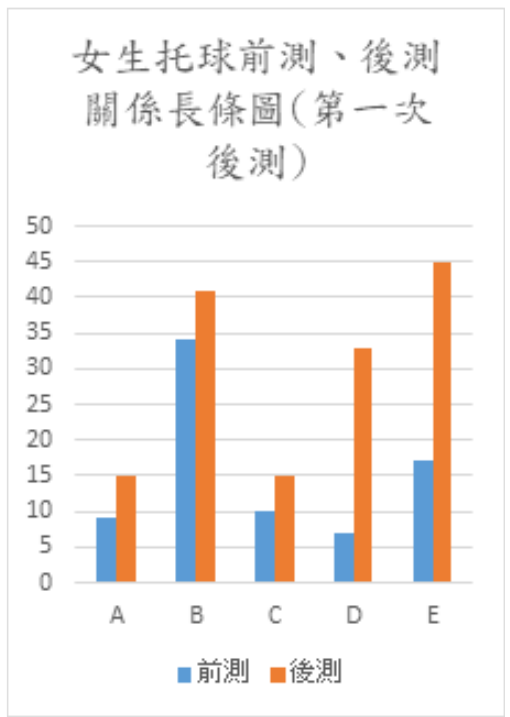
圖四十一，男生托球前、後測長條圖
(第一次後測)(第二作者製表)



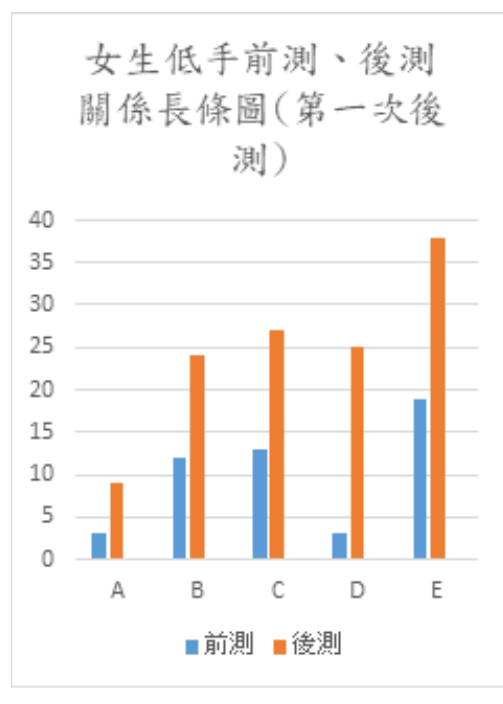
圖四十二，男生低手前、後測長條圖
(第一次後測)(第二作者製表)



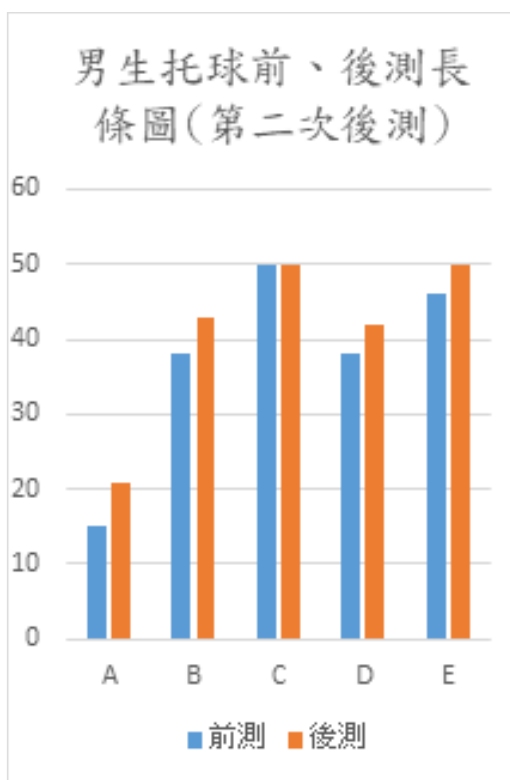
圖四十三，女生托球前、後測長條圖(第一次後測)(第二作者製表)



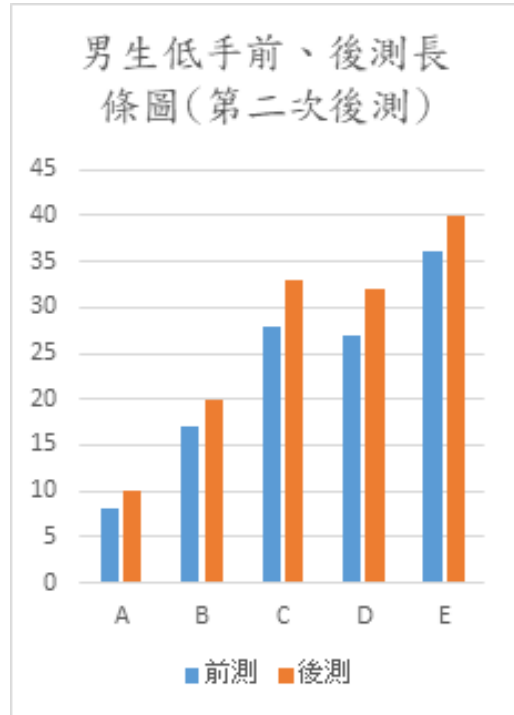
圖四十四，女生低手前、後測長條圖(第一次後測)(第二作者製表)



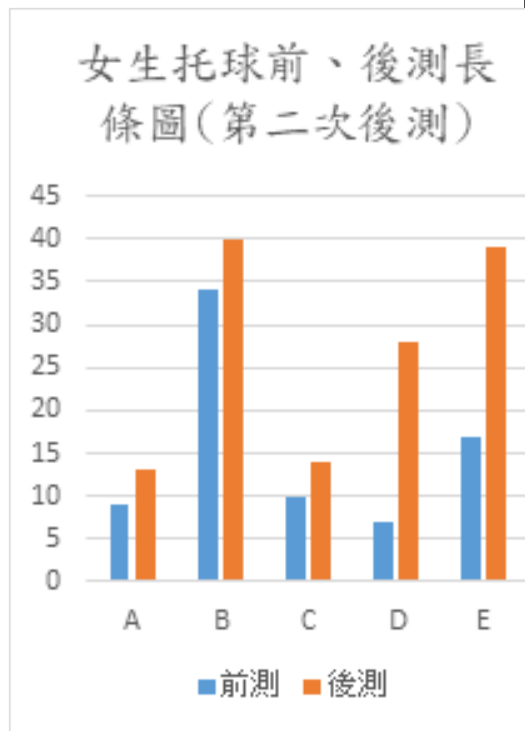
圖四十五，男生托球前、後測長條圖(第二次後測)(第二作者製表)



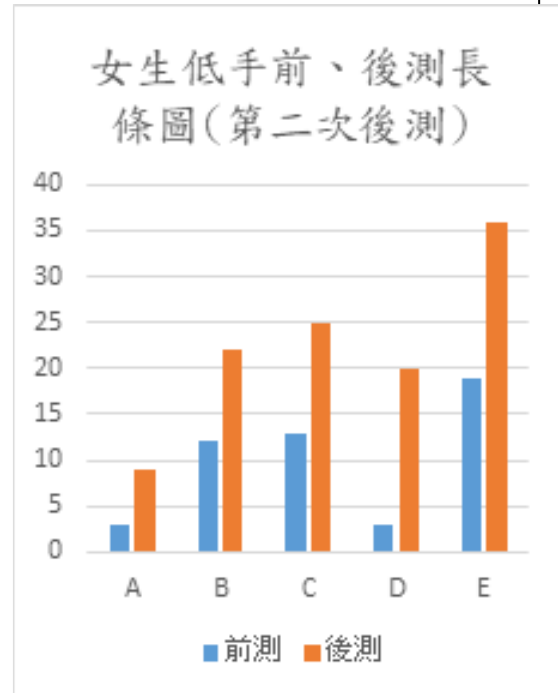
圖四十六，男生低手前、後測長條圖(第二次後測)(第二作者製表)



圖四十七，女生托球前、後測長條圖
(第二次後測)(第二作者製表)



圖四十八，女生低手前、後測長條圖
(第二次後測)(第二作者製表)



伍、討論

- 一、在做影像辨識時，拍攝者最好穿緊身的球衣，以免衣服太寬鬆，影響關節位置的判斷。
- 二、用 mediapipe 來分析動作非常方便，不必再做機器的訓練模式，只要寫好程式即可上手，只要有國中數學的程度，就可以完成。
- 三、本次研究用的軟體如 python、opencv、mediapipe、arduino，都是免費的，可以在經費有限的情況下，做最大的發揮。
- 四、本研究針對的是基礎的排球動作，接球都是正面回擊到正前方，沒有傳球到特定角度，如此處理起來較為單純。
- 五、每位排球選手在擊球前的準備動作都不盡相同，但是擊球的瞬間動作一致，所以分析動作時，只擷取擊球瞬間的照片即可。

六、由於打排球的時候，人的移動不會太快，所以用一般的攝影機就可以，不會有殘影出現。除非是要分析扣球的動作，才用到高速攝影。所以對於基礎運動員來說，一般的設備就足夠了。

陸、結論

- 一、mediapipe 軟體是由 Google Research 所開發的多媒體機器學習模型，該公司已經針對不同種族、年齡的人檢測，可以充分辨識人的姿勢、表情，所以藉由 mediapipe 我們可以輕鬆地跨入影像辨識的領域。
- 二、計算角度需用到三角函數，而程式碼算出的角度是弧度，也就一個圓的角度是 2π ，與我們熟悉的一個圓 360 度不一樣，所以要必須經過數學換算，才能變成我們熟悉的角度。
- 三、在受測者的運動服不太寬鬆的情況下，AI 排球教練均能辨識出受測者各個關節的角度，誤差不會太大，所以用 AI 教練來判斷受測者的動作是否標準可行。但是對於細部的托球動作，失之毫釐，差之千里，無法用影像辨識的方法來判斷手指的動作，所以用壓力感測器配合 ARDUINO 來顯示各個手指觸球的時間，可以精確的判斷各個手指的觸球時間差距，作為運動員修正的參考。
- 四、在科技日新月異的時代，運動員的訓練方式已經科技化，但排球運動的訓練模式仍是一層不變，都是由教練用眼睛看，在指導學生修正錯誤。這種方法沿用了數十年，教練一次只能指導一個學生，效率較低，學生也因為看不到自己的動作與標準動作差哪裡，修正起來與教練的要求有落差。藉由 AI 教練，運動員可以有客製化的學習方式，可以看到自己的缺點，也可以了解到修正的方式，效果強，效率高，值得來推廣。
- 五、未來 AI 排球教練還可以分析舉球員的表情，判斷是要舉出哪一種戰術，也可以判斷攻擊手的動作或表情，是真正要攻擊還是虛晃一招，藉由這些功能的延伸，可以提升排球的水準，讓 AI 排球教練由動作矯正擴展到戰術判斷，變成功能全面化的 AI 教練。
- 六、經過前測與後測的比較，發現大部分的選手在後測後成績都有顯著的提升，可見及時矯正受測者的動作對於受測者學習排球的成效來說是有幫助的。所以利用 AI 排球教練的軟體可以有效的提升受測者學習排球的效果，而且可以節省人力，是一種非常方便的輔助教學器材，值得推廣。

柒、參考文獻資料

- 一、STEAM 教育學習網。
- 二、鄧文淵，文淵閣工作室，Python 初學特訓班(第三版):從快速入門到主流應用全面實戰，2019 年 12 月三版，碁峰資訊股份有限公司
- 三、Martin Evans，Joshua Noble，Jordan Hochenbaum 作，王冠勛譯，Arduino 完全實戰手冊，2014 年 2 月，博碩文化股份有限公司
- 四、楊明豐，Arduino 最佳入門與應用-----打造互動設計輕鬆學 2014 年 5 月初版二刷，碁峰資訊有限公司

【評語】 032803

1. 本作品以 AI 影像辨識分析排球球員的動作，構想清晰，並提供球員改進的建議，成功結合運動與科技，為運動訓練提供了新的途徑，表現優良，值得讚賞。
2. 進行實驗時，選取不同程度的球員（好、中、差）進行測試，方法符合科學實驗精神。
3. 目前的作品，顯示後測比前測有明顯進步。若能將真人指導的進步幅度納入對照組進行比較，將可使作品更臻完善。

作品簡報

AI排球教練

摘要

現在流行AI，我們就想設計一款程式來教人打排球。首先下載python、opencv和mediapipe的Pose landmark detection guide，然後寫入程式，印出關節點的座標。接著找排球隊員，拍攝托球、低手接球、發球、扣球的影片，然後擷取碰球瞬間的影格，載入程式中並計算各個關節點之間的角度，從而定義出標準動作中的各個角度。接著拍攝初學者的各種基本動作影片，然後依照同樣的方法計算各關節點的角度，再與標準版的角度做比較，即可列出初學者在各個關節點應該修正的角度。再利用Arduino及壓力傳感器做出托球矯正手套，讓人托球之後用LED顯示各個指頭觸球的先後順序，再讓人矯正手指的動作。藉由AI的操作，可以讓初學者快速知道自己的缺點，也知道修正的方法，這就是AI教練的魅力。

壹、前言

一、研究動機

學校在上排球體育課的時候，老師在教同學做一些基本動作，明明老師都指證了同學的動作錯在哪裡，但是同學在下次再做相同動作的時候，依然會做錯，讓老師不斷的糾正到很煩，於是我們就想到了資訊課所教的程式設計，想用影像辨識的方式來找出同學錯誤的地方以及修正的參考，期望同學對攝影機打球就會顯示自己的錯誤之處，並顯示改正的方法。

二、研究目的

- (一) 找出可以做影像辨識的程式
- (二) 找出python程式中與影像辨識有關的指令
- (三) 找出opencv與影像相關的程式碼
- (四) 用python與opencv顯示出影像
- (五) 下載mediapipe程式，顯示各個關節的座標點
- (六) 寫出關節彎曲角度的程式
- (七) 找出排球的各種基本動作，並找動作最標準的人拍成影片
- (八) 找出各種標準動作各關節的彎曲角度
- (九) 在初學者的影片中顯示出與標準動作的角度差
- (十) 分析初學者的錯誤地方及矯正方式
- (十一) 測量托球六個手指的碰球時間
- (十二) 製作低手傳球感測袖套
- (十三) 利用前測與後測比較研究的成果

三、文獻回顧

- (一)、中華民國第63屆中小學科學展覽會 國小組生活與應用科學科 作品名稱:挺立-應用全身辨識輔助站姿調整即時回饋系列之研究,該作品是用機器學習的方式來判斷角度的變化,所以辨識度會有一些誤差。
- (二)、中華民國第 61 屆中小學科學展覽會 國中組 生活與應用科學(一) 作品名稱:搶救生命大作戰- AI 姿態辨識在智慧型高品質 CPR 訓練引導式教學輔具系統設計之研究 該作品是利用各種感測器來偵測操作者的動作是否正確。
- (三)、中華民國第 60 屆中小學科學展覽會 高級中等學校組 工程學(一)科 作品名稱:基於 OpenPose 之機具操作於異常姿勢即時偵測,該作品是利用影像辨識出各個關節點的座標,在利用高中的數學來計算角度,從而判斷姿勢是否正確。

貳、研究設備及器材

一、Arduino 如圖一



二、壓力感測器 如圖二



三、AI智慧手套正、反面 如圖三、四



四、AI智慧袖套正面,如圖五



參、研究過程和方法

一、找出可以做影像辨識的程式

1. 上網搜尋跟影像辨識有關的程式
2. 篩選適合程度的軟體並下載
3. 實際測試可行的程度
4. 選擇適用的軟體

二、找出python程式中與影像辨識有關的指令

1. 下載python程式
2. 找出計算距離和角度的相關程式碼
3. 練習計算點與點之間的距離以及兩條線的角度

三、找出opencv與影像相關的程式碼

1. 下載opencv程式
2. 建立一個空畫面
3. 找出與繪圖有關的程式碼並執行
4. 找出在圖片寫文字的程式碼並執行

四、用python與opencv顯示出影像

1. 在opencv程式中載入一個影像檔
2. 在影像上面繪圖與寫字

五、下載mediapipe程式，顯示各個關節的座標點

1. 到mediapipe網站下載有關於全身座標的程式。
2. 適當修改程式碼印出特定座標的程式

六、寫出關節彎曲角度的程式

1. 找出低手接球相關座標點
2. 利用三角函數計算關節彎曲的角度

七、找出排球的各種基本動作，並找動作最標準的人拍成影片

1. 詢問排球教練有哪些基本動作
2. 請排球教練挑一個動作最標準的選手來拍影片

八、找出各種標準動作的關節彎曲角度

1. 從拍攝的影片中擷取碰球瞬間的照片
2. 將截圖送至python檔案並計算各關節的彎曲角度

九、在初學者的影片中顯示出與標準動作的角度差

1. 找一位初學者來拍攝各種排球動作的影片
2. 擷取碰球瞬間的照片
3. 將截圖送至python檔案並計算各關節的彎曲角度

十、分析初學者的錯誤地方及矯正方式

1. 較初學者與標準版的關節彎曲角度差
2. 告知初學者需要修正的角度
3. 請初學者練習幾次後再拍成影片直到完全校正為止

十一、測量托球六個手指的碰球時間

1. 在手套上縫製壓力感測器及LED燈泡
2. 連結至arduino。
3. 寫程式設定當觸球的時候，紀錄那根手指的觸球時間。
4. 觸球後，延遲3秒，讓燈光發亮6秒。
5. 利用燈泡亮起的時間比較觸球時間是否相同

十二、製作低手傳球感測袖套

1. 把壓力感測器一端連接5v、一端連接電阻及GND，然後再連接到A0作為右手的感測器。左手則連接到A1，作為左手的感測器。
2. 將壓力感測器用塑膠套包起來後，縫在袖套上，縫製的地方要對準擊球點。
3. 將mp3播放模組的接線連接到Arduino
4. 寫入程式碼，當兩個擊球點碰撞的時間在0.01秒以內就發布正確的訊號。如果右手的時間比左手的時間晚了0.01秒以上，就發布右手高一點的訊號，反之則發布右手低一點的訊號。

十三、利用前測與後測比較研究的成果

1. 前測 2. 後測 3. 比較

肆、研究結果

一、找出可以做影像辨識的程式

(一) 上網搜尋跟影像辨識有關的程式

1. 機器學習範例程式說明
此方法主要是利用大量的圖片讓機器學習從而做出正確的影像判斷，但是對於同樣的姿勢只有細微差別的動作卻是無法分辨，所以不適合本實驗的需求。

2. Steam教育學習網

此方法是用python軟體，載入opencv、mediapipe程式，可以列出關節點的座標，然後自己再利用數學公式計算出關節點的角度，很適合本實驗的需求，故採用本方法

(二) 篩選適合程度的軟體並下載

1. 下載python程式
(1) 到python官網，選擇 [Windows installer \(64-bit\)](#) 下載到電腦裡面。
(2) 開啟下載的程式並安裝，安裝過程中要勾選path選項。
2. 下載visual studio code程式
(1) 到visual studio 官網，選擇Download for windows，下載完成後安裝
3. 下載opencv程式
打開visual studio code程式，在終端機畫面鍵入pip install opencv-python即完成
4. 下載mediapipe程式
打開visual studio code程式，在終端機畫面鍵入pip install mediapipe即完成。接著到mediapipe的網站，下載姿勢的程式碼如附件。

四、用python與opencv顯示出影像

- (一) 在opencv程式中載入一個g1影像檔並在影像上面寫字，程式碼如下

```
img1=cv2.imread('g1.png')
text = '(161, 200, 249)'
cv2.putText(img, text, (10, 100), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 255, 0), 5, cv2.LINE_AA) #如圖九所示。
```

二、找出python程式中與影像辨識有關的指令

(一) 找出計算距離和角度的相關程式碼

1. 下載math程式庫
2. 選擇常用的數學函數如下:
(1) a^b 的表示法: `math.pow(a,b)`。
(2) `sina=math.sin(a)`
(3) `cosa=math.cos(a)`
(4) `tana=math.tan(a)`

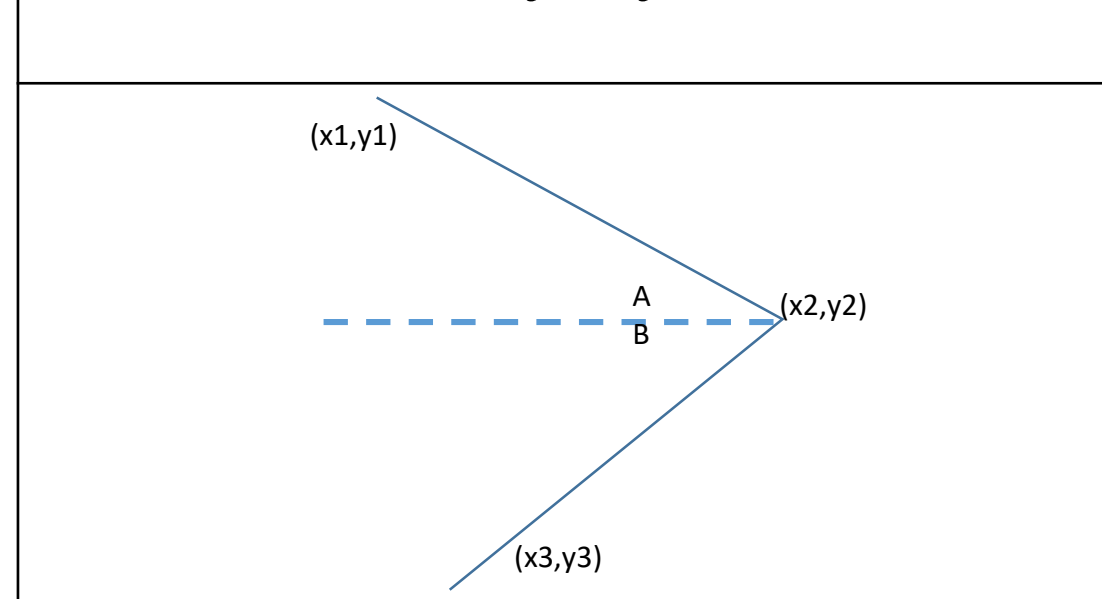
(二) 練習計算點與點之間的距離

1. 兩點的座標設為 $(x1,y1)$ ， $(x2,y2)$
2. 這兩點的距離數學方程式為
$$\sqrt{(x1-x2)^2+(y1-y2)^2}$$
3. 把式子寫成python程式為
`(x1-x2)2=math.pow((x1-x2, 2))=x`
`(y1-y2)2=math.pow((y1-y2, 2))=y`
兩點的距離=`math.pow((x+y), 0.5)`

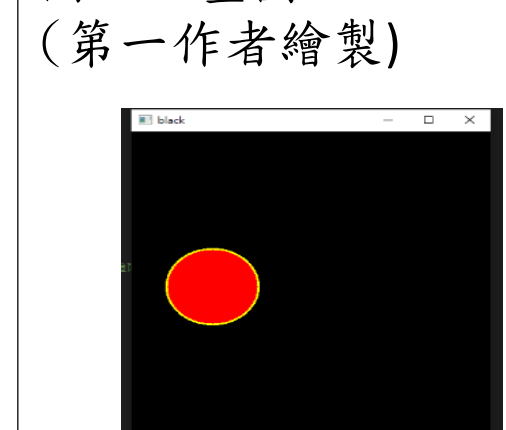
(三) 計算兩條線的夾角

1. 假設兩條線的三個點的座標分別為 $(x1,y1)$ ， $(x2,y2)$ ， $(x3,y3)$ 如圖六所示

圖六 計算兩條線的夾角示意圖
 $A=\text{math.atan}((y2-y1)/(x2-x1))$
 $B=\text{math.atan}((y3-y2)/(x3-x2))$



圖七: 畫圖 (第一作者繪製)

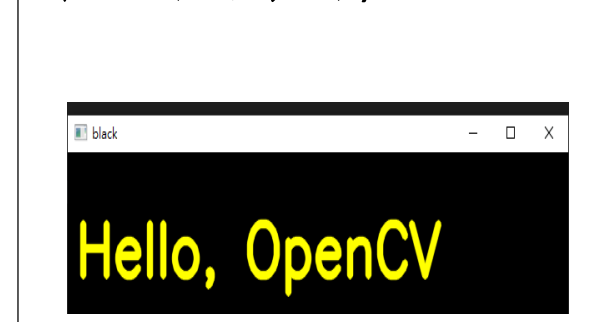


三、找出opencv與影像相關的程式碼

- (一) 下載opencv程式
- (二) 建立一個空畫面
- (三) 找出與繪圖有關的程式碼並執行
- (四) 找出圖片寫文字的程式碼並執行

```
import numpy as np
import cv2
# 開一個空畫面:高600,寬300
img = np.zeros((600, 300, 3), np.uint8)
#cv2.circle(影像, 圓心座標, 半徑, 顏色, 線條寬度)如圖七所示。
cv2.circle(img, (90, 210), 50, (0, 255, 255), 3)
text = 'Hello, OpenCV'
# 使用各種字體 cv2.putText(影像, 文字, 第一字左下角坐標, 字形, 大小, 顏色, 寬度) 如圖八所示
cv2.putText(img, text, (10, 100), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 255, 255), 5, cv2.LINE_AA)
```

圖八: 顯示文字示意圖 (第一作者製作)



圖九: 正面接球 (第一作者拍攝)



五、下載mediapipe程式，顯示各個關節的座標點

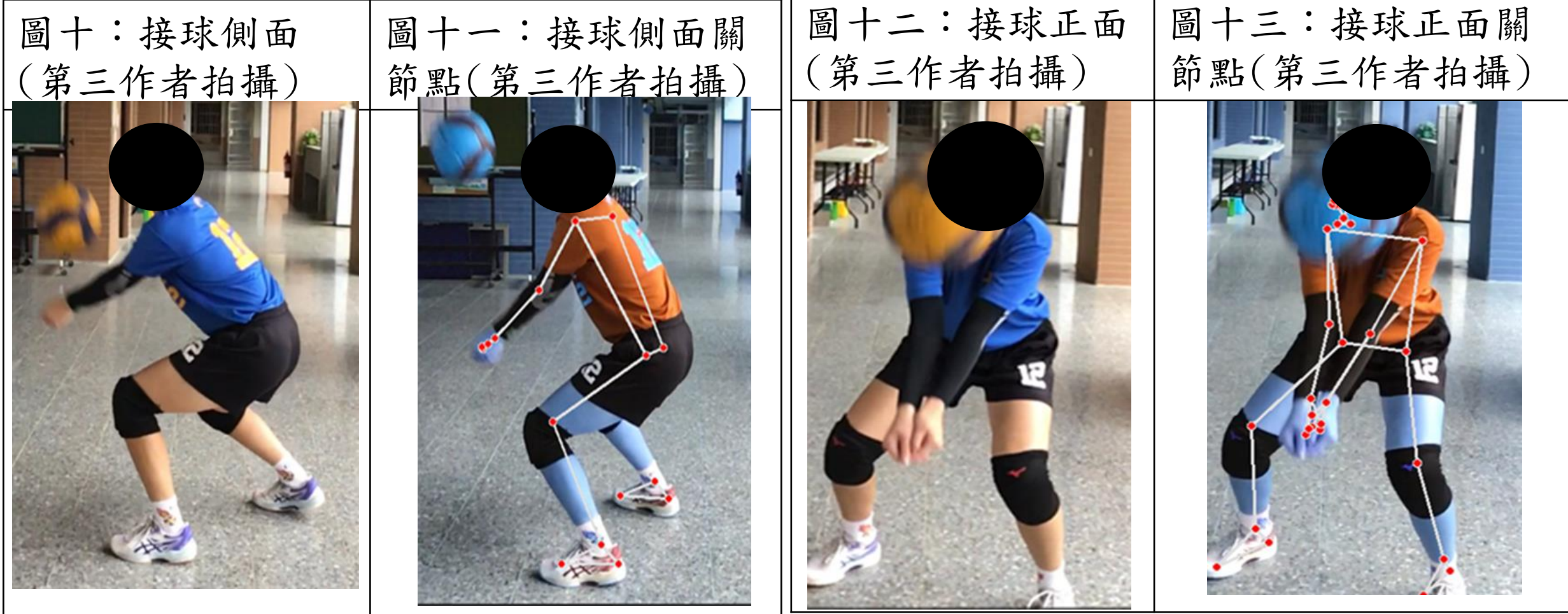
(一)到mediapipe網站下載有關於全身座標的程式

```
import cv2
import mediapipe as mp
mp_drawing = mp.solutions.drawing_utils # mediapipe 繪圖方法
mp_drawing_styles = mp.solutions.drawing_styles # mediapipe 繪圖樣式
mp_holistic = mp.solutions.holistic # mediapipe 全身偵測方法
cap = cv2.VideoCapture(0)
# mediapipe 啟用偵測全身
with mp_holistic.Holistic(
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as holistic:
    if not cap.isOpened():
        print("Cannot open camera")
        exit()
    while True:
        ret, img = cap.read()
        if not ret:
            print("Cannot receive frame")
            break
        img = cv2.resize(img, (520, 300))
        img2 = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # 將 BGR 轉
        # 換成 RGB
        results = holistic.process(img2) # 開始偵測全身
        # 面部偵測，繪製臉部網格
        mp_drawing.draw_landmarks(
            img,
            results.face_landmarks,
            mp_holistic.FACEMESH_CONTOURS,
            landmark_drawing_spec=None,
            connection_drawing_spec=mp_drawing_styles
            .get_default_face_mesh_contours_style())
        # 身體偵測，繪製身體骨架
        mp_drawing.draw_landmarks(
            img,
            results.pose_landmarks,
            mp_holistic.POSE_CONNECTIONS,
            landmark_drawing_spec=mp_drawing_styles
            .get_default_pose_landmarks_style())
        cv2.imshow('oxxostudio', img)
        if cv2.waitKey(5) == ord('q'):
            break # 按下 q 鍵停止
    cap.release()
    cv2.destroyAllWindows()
```

(二)適當修改程式碼印出特定座標的程式

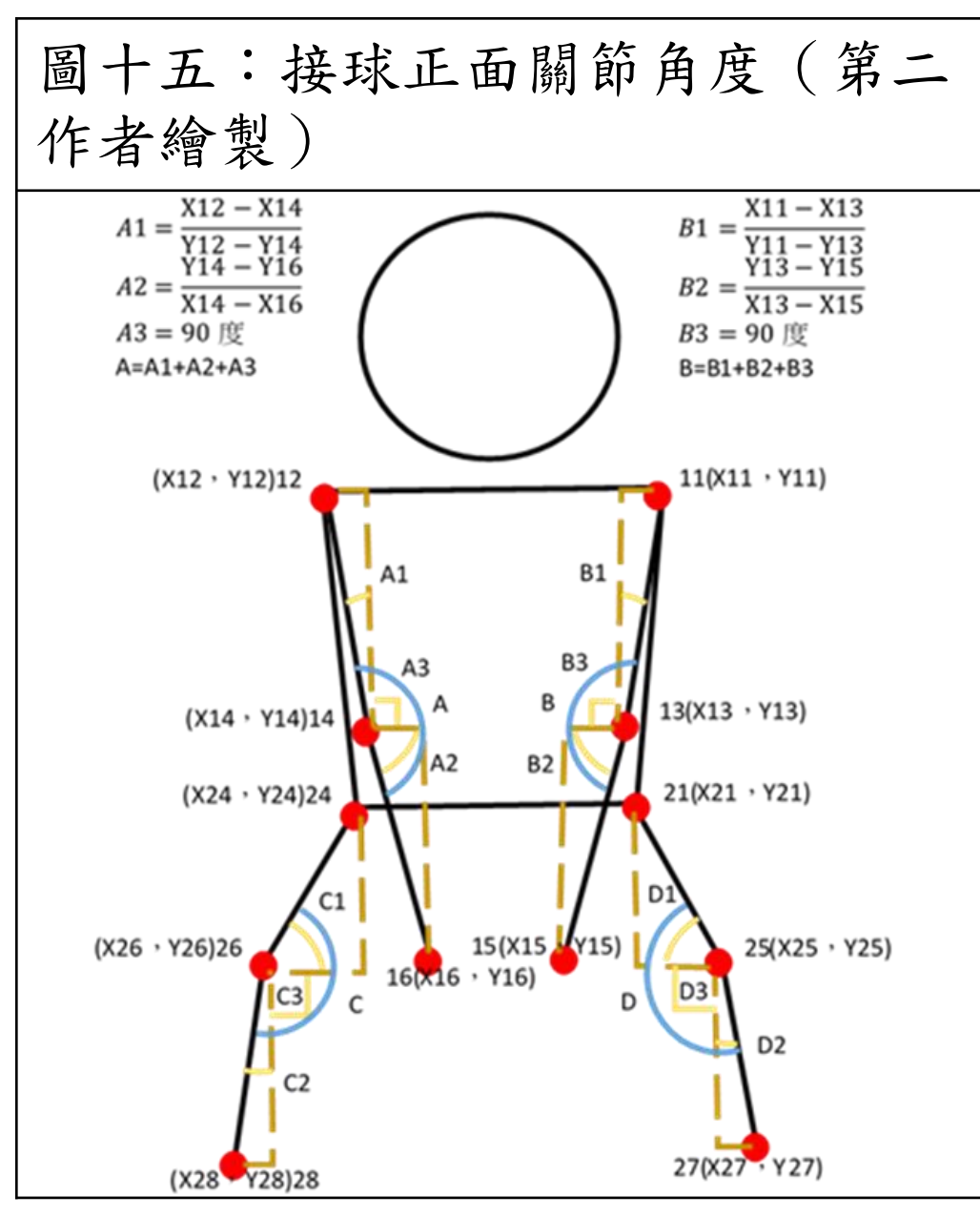
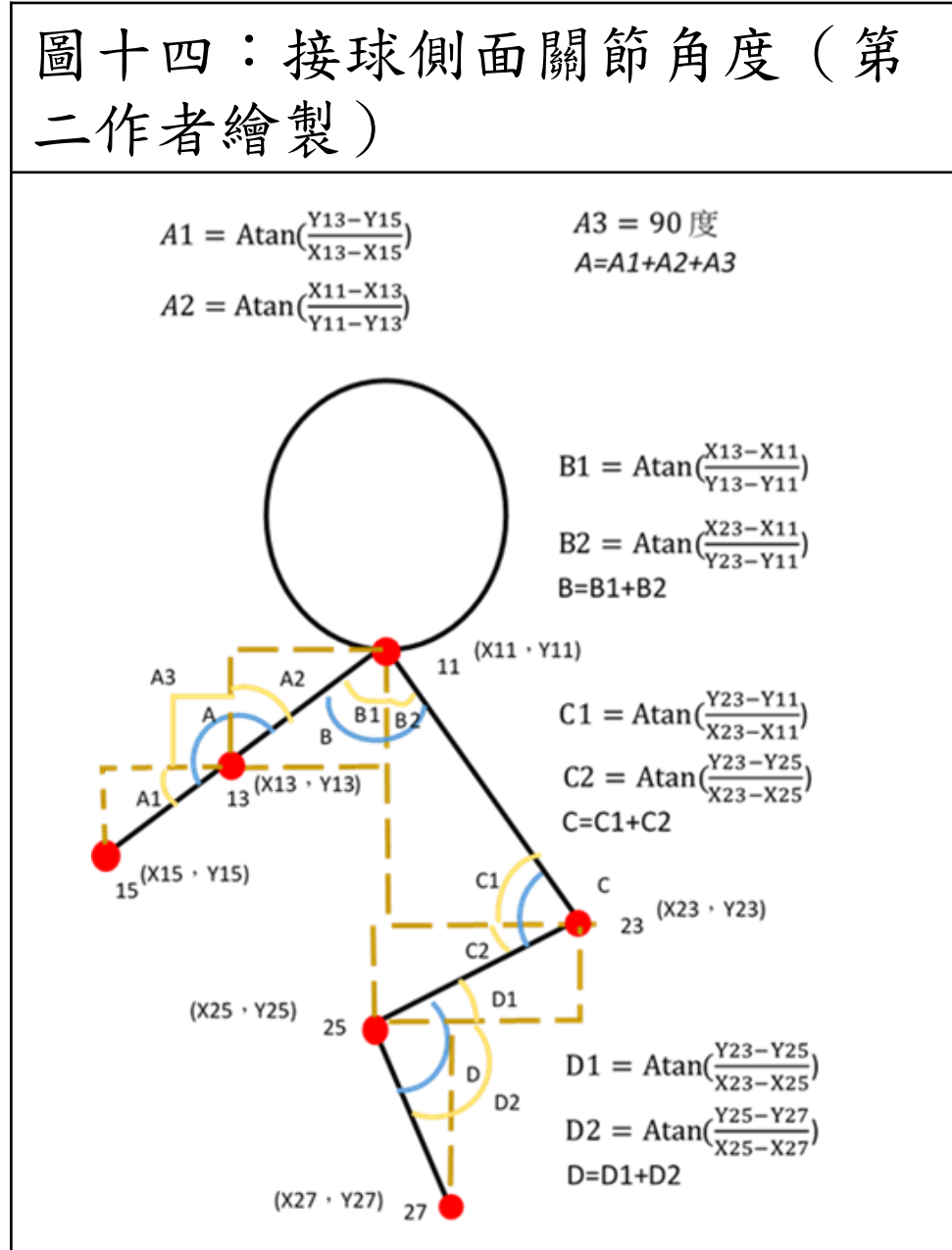
六、寫出關節彎曲角度的程式

(一)找出低手接球相關座標點



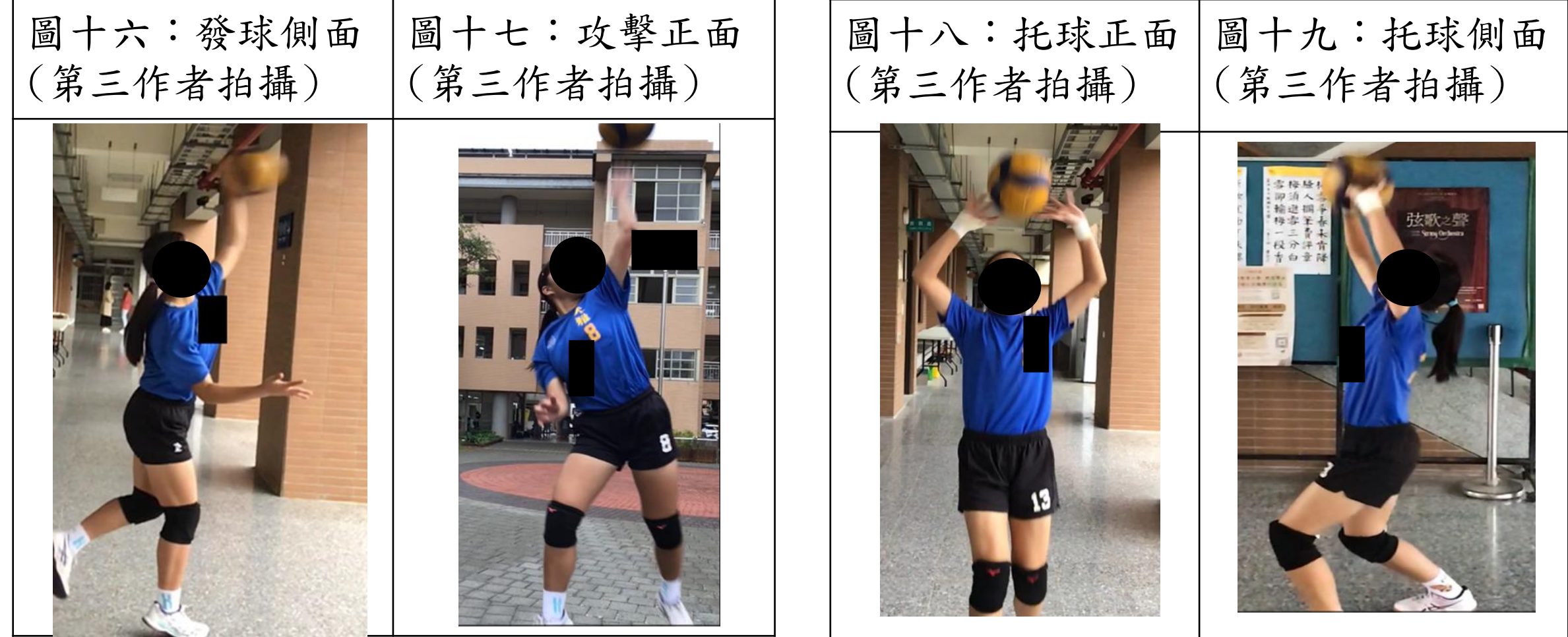
(二)利用三角函數計算關節彎曲的角度

1. 接球正面角度計算 2. 接球側面角度



七、找出排球的各種基本動作，並找動作最標準的人拍成影片

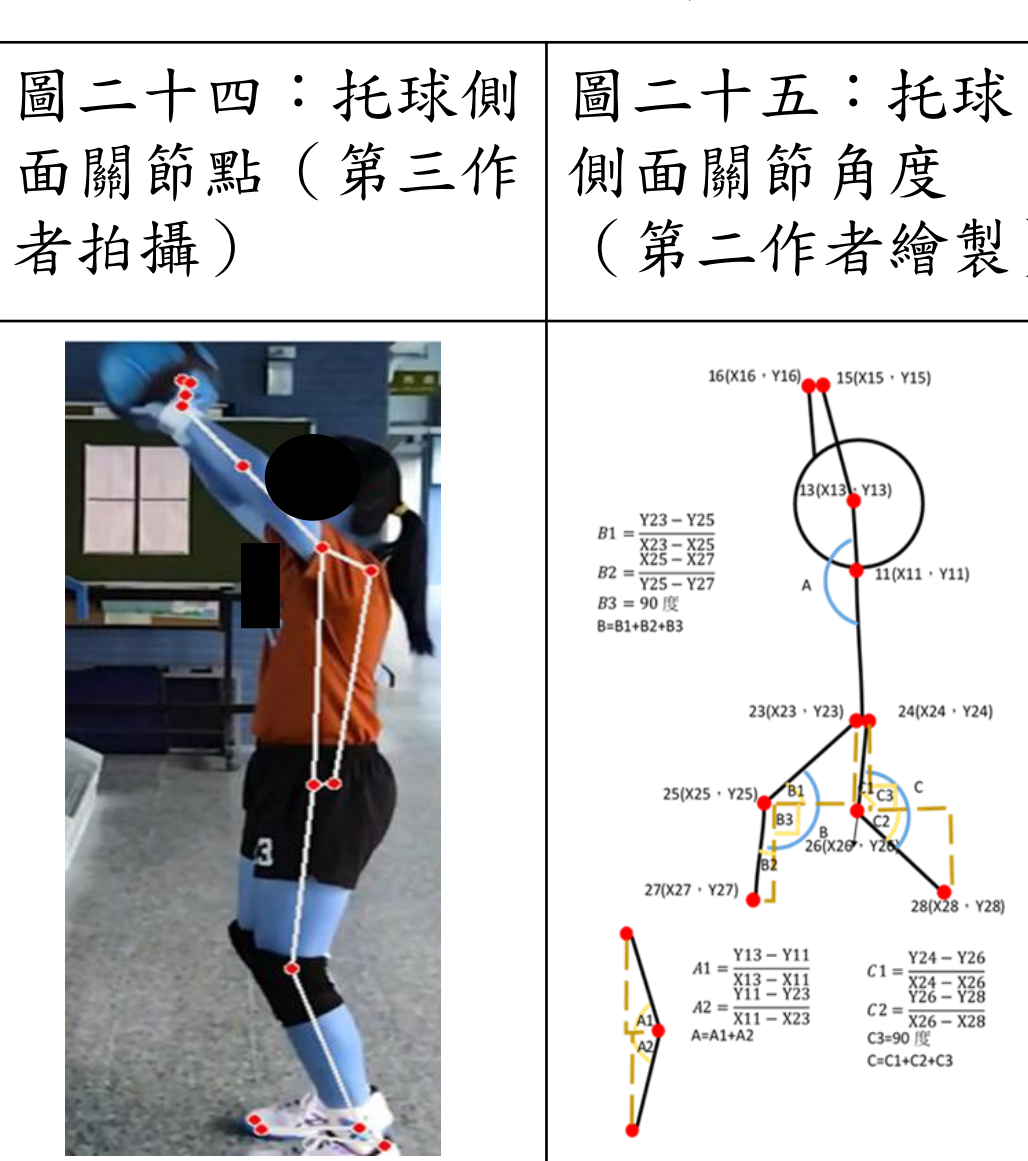
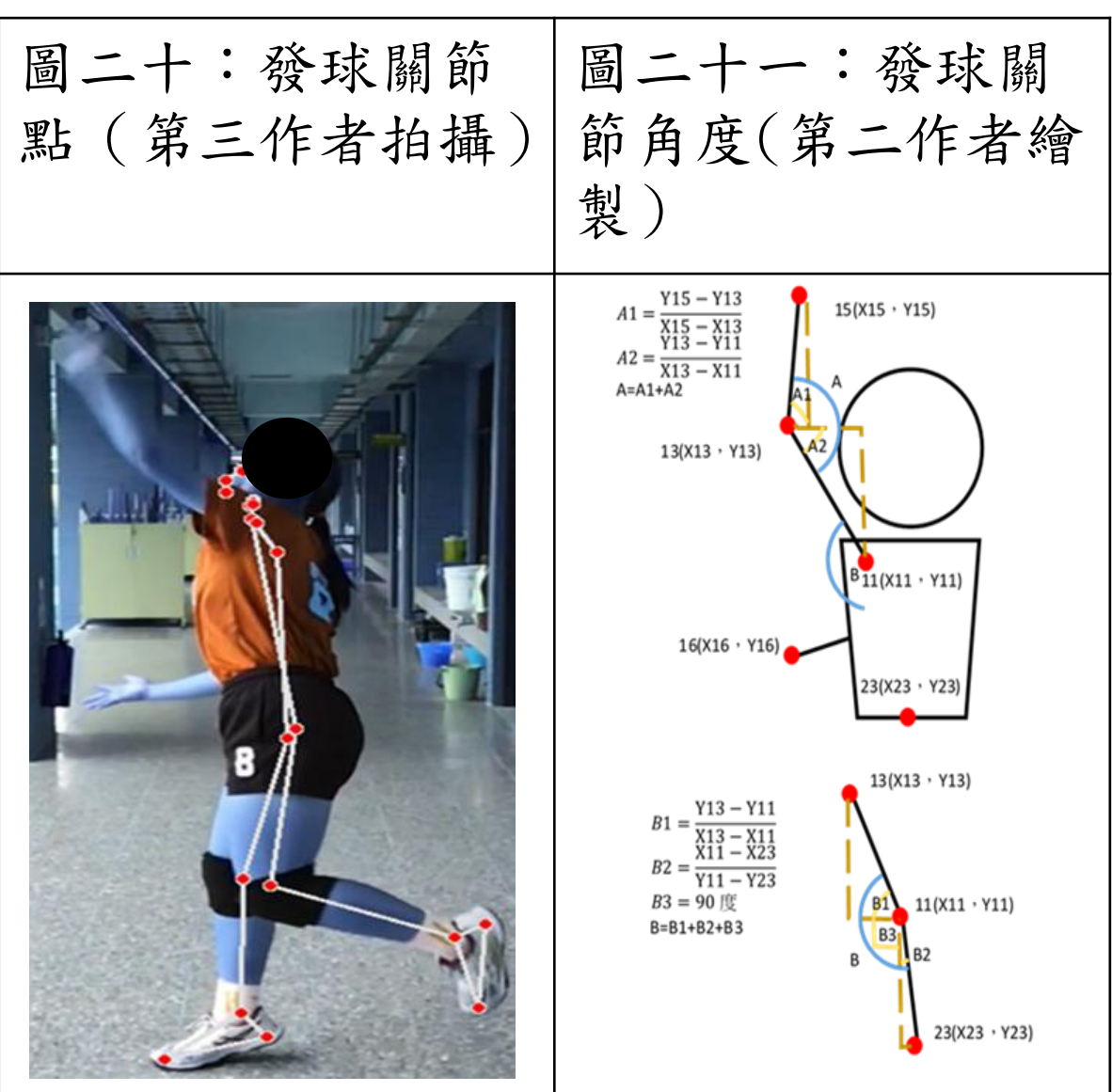
(一)拍攝排球基本動作



八、找出各種標準動作的關節彎曲角度

(一)發球角度計算

(二)托球側面角度計算

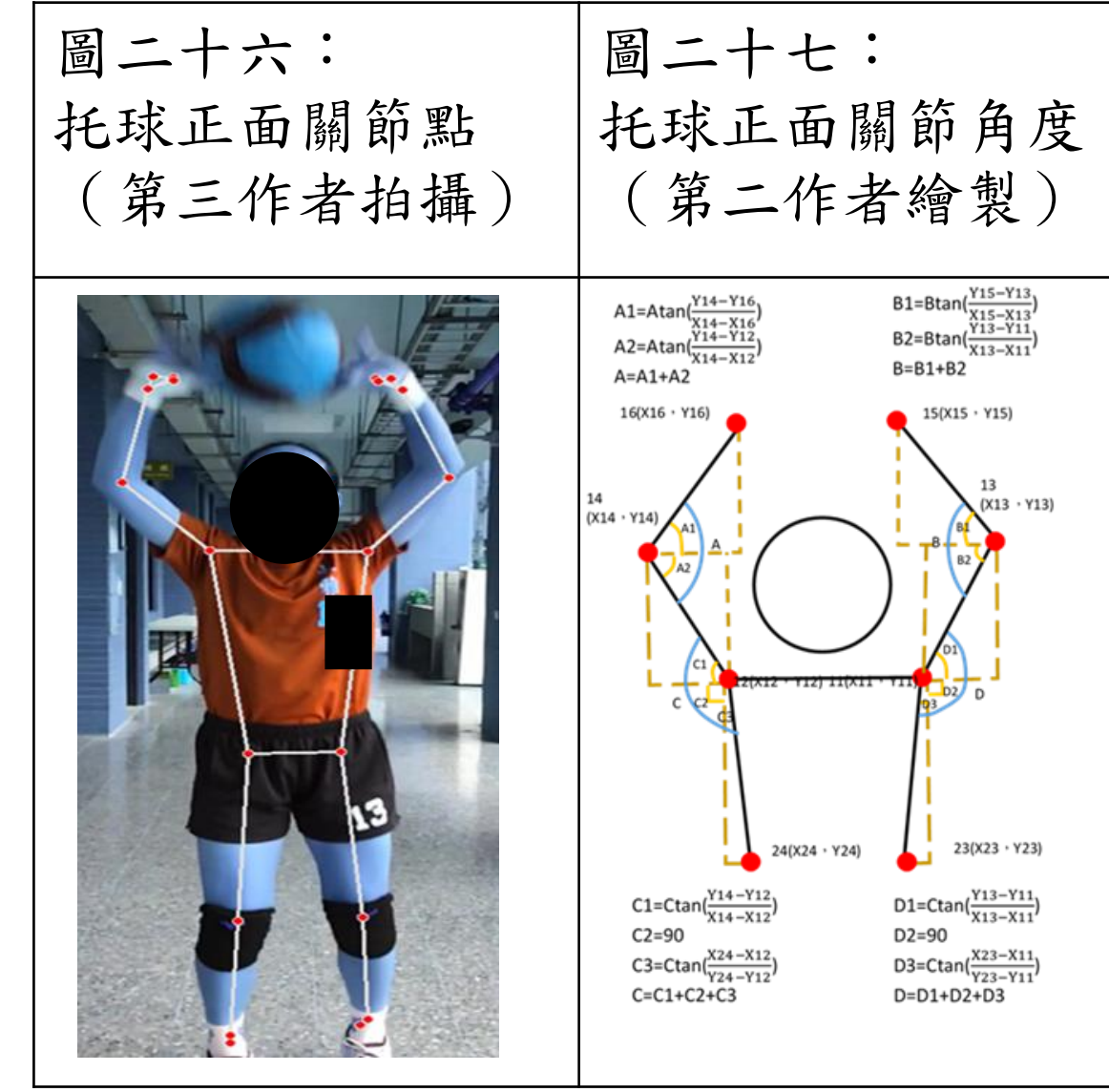
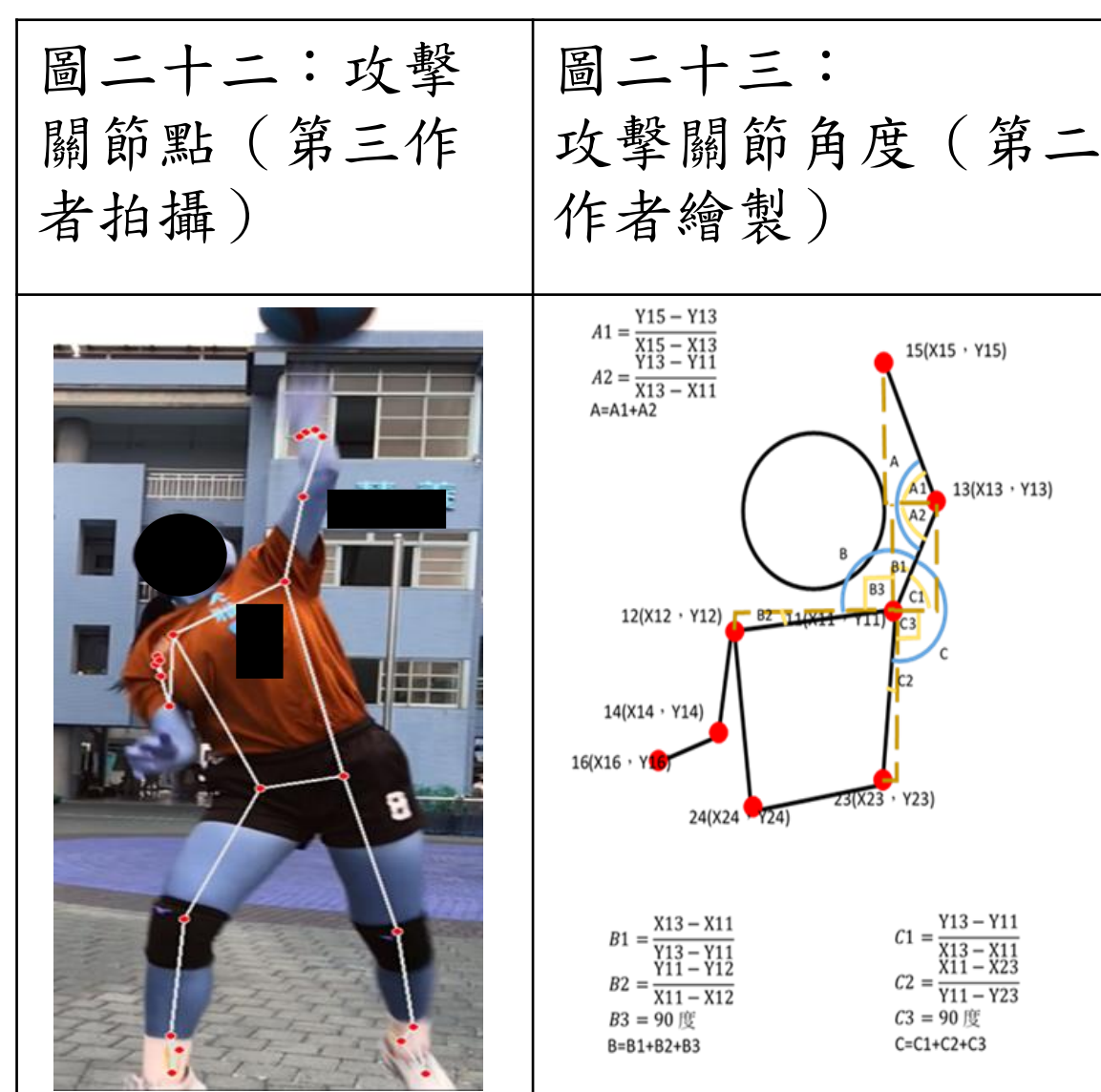


(二)發球側面程式碼(詳見附錄三)

(三)攻擊角度計算

(六)托球側面程式碼(詳見附錄五)

(七)托球正面角度計算



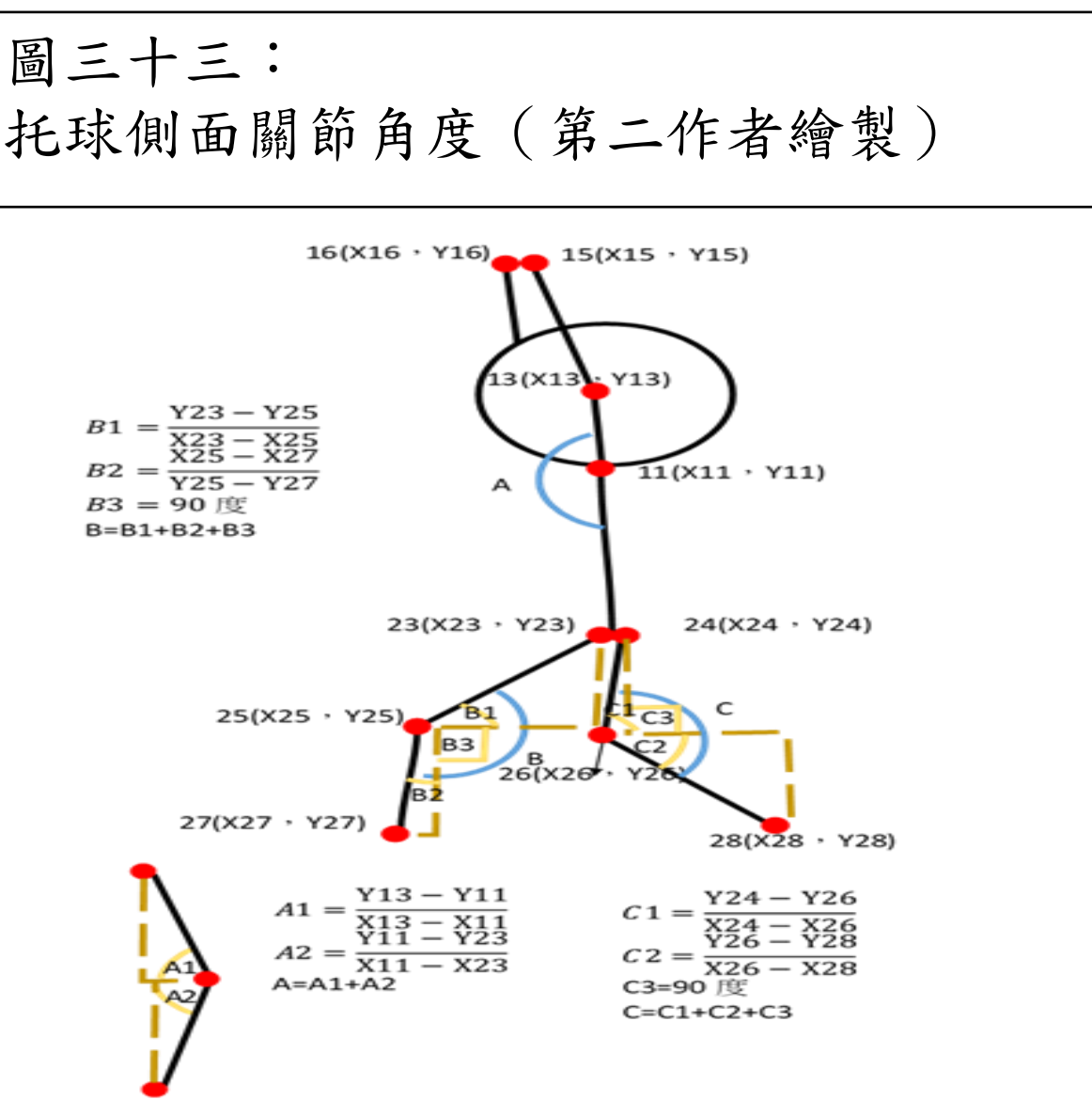
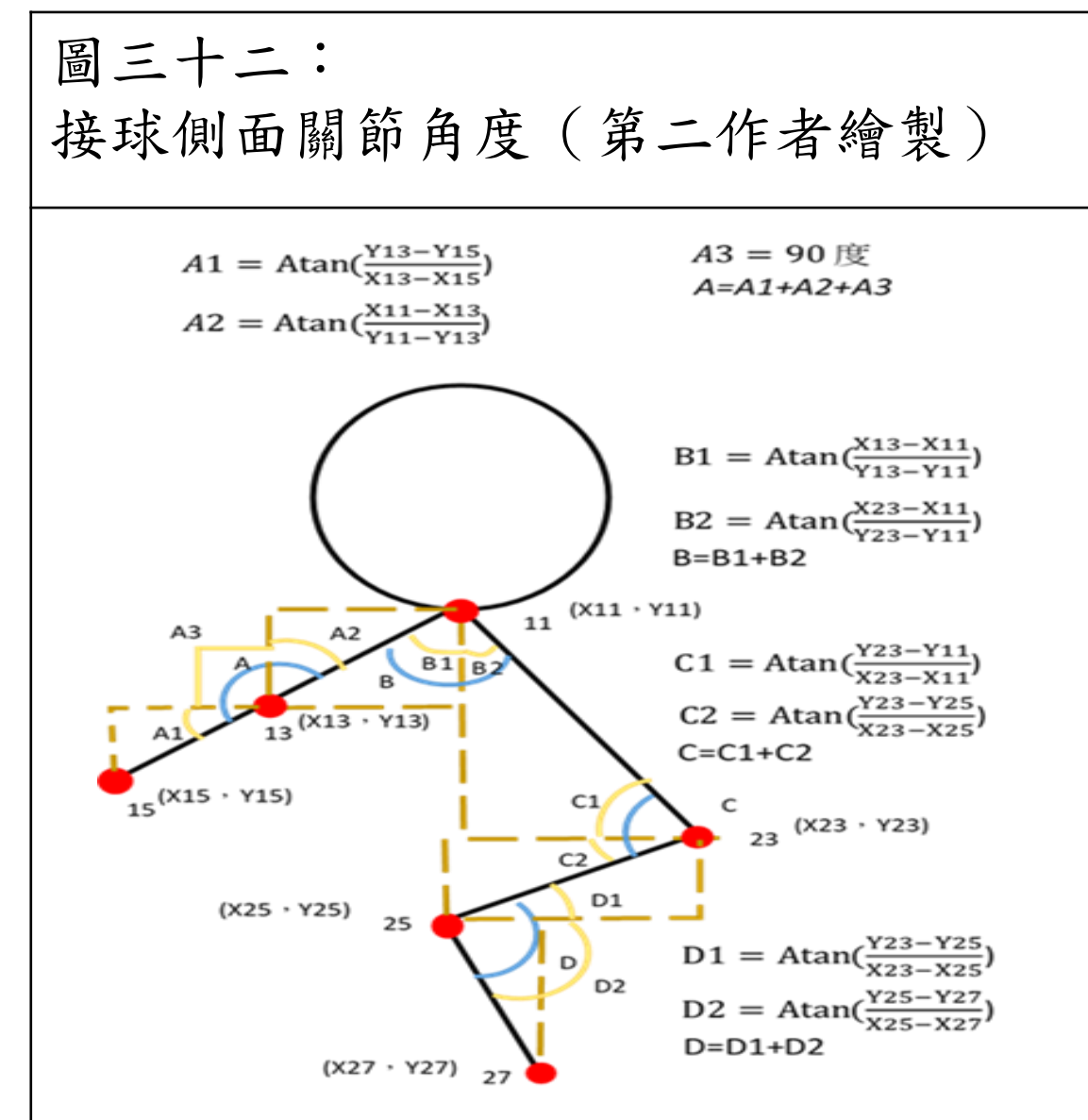
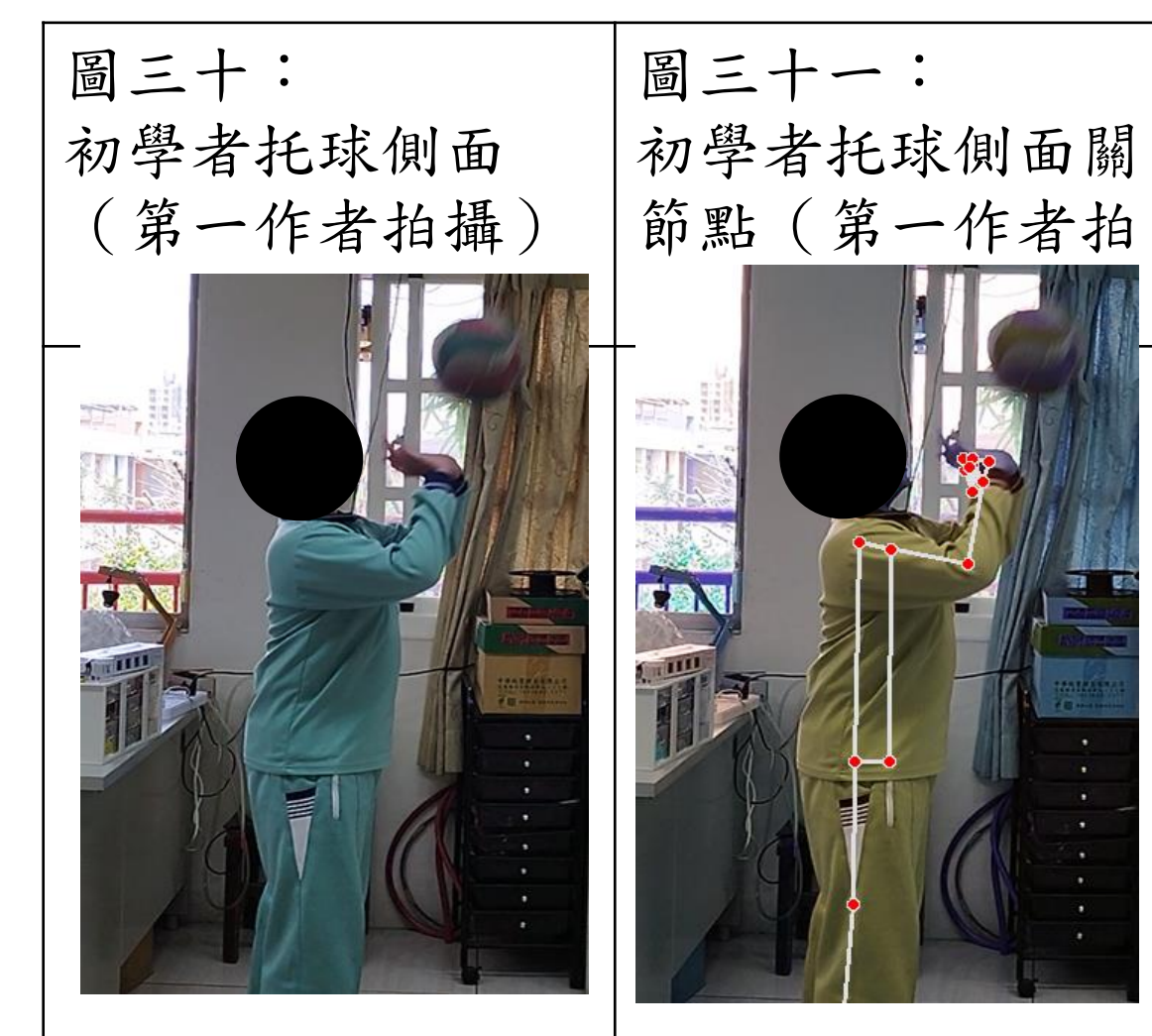
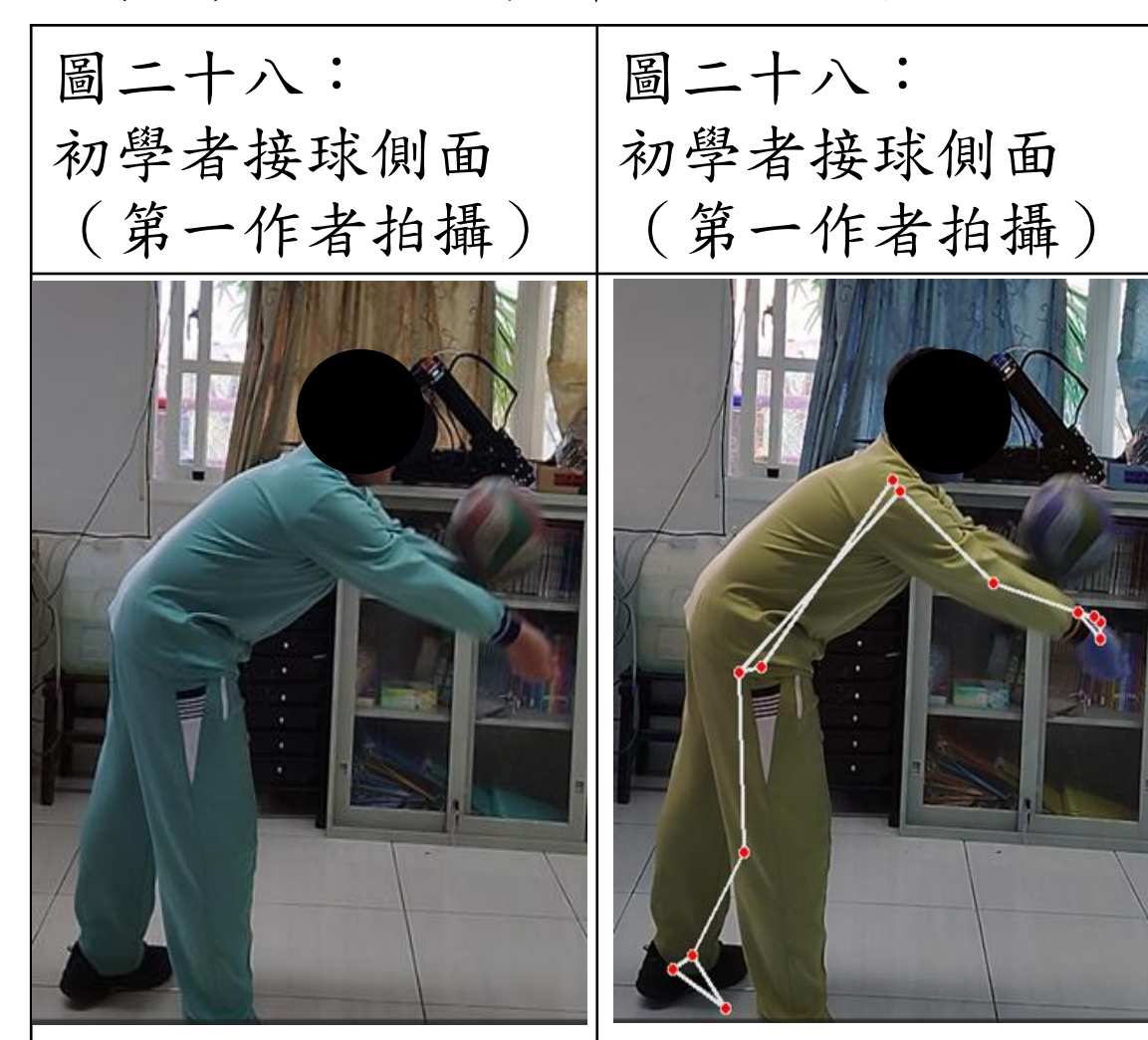
(四)攻擊正面程式碼(詳見附錄四)

(八)托球側面程式碼(詳見附錄六)

九、在初學者的影片中顯示出與標準動作的角度差

(一)找一位初學者來拍攝各種排球動作的影片

(二)擷取碰球瞬間的照片



十、分析初學者的錯誤地方及矯正方式

(一)比較初學者與標準版的關節彎曲角度差，如表一、表二所示。

表一: 低手接球初學者與標準版的關節彎曲角度差

	角A	角B	角C	角D
側面接球標準版	164度	129度	92度	98度
側面接球初學者	142度	138度	137度	152度

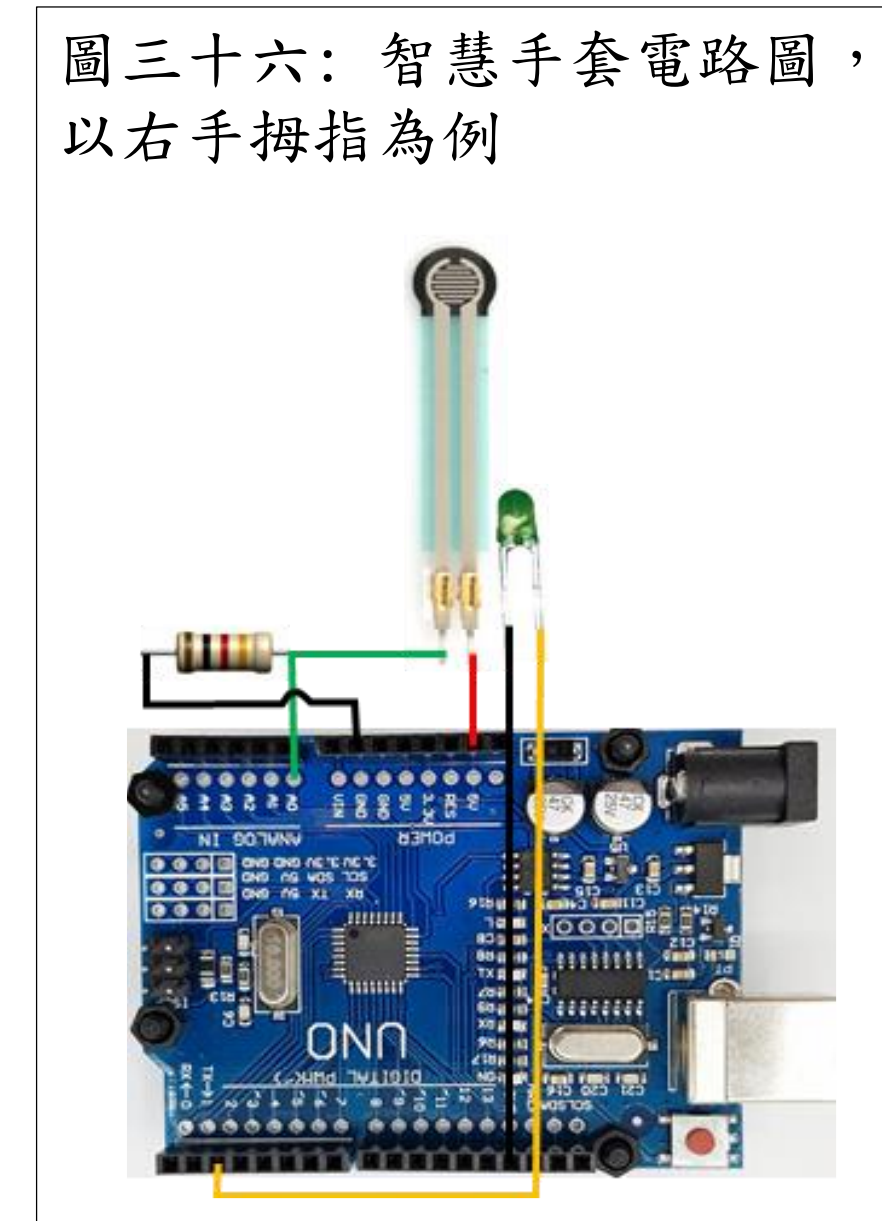
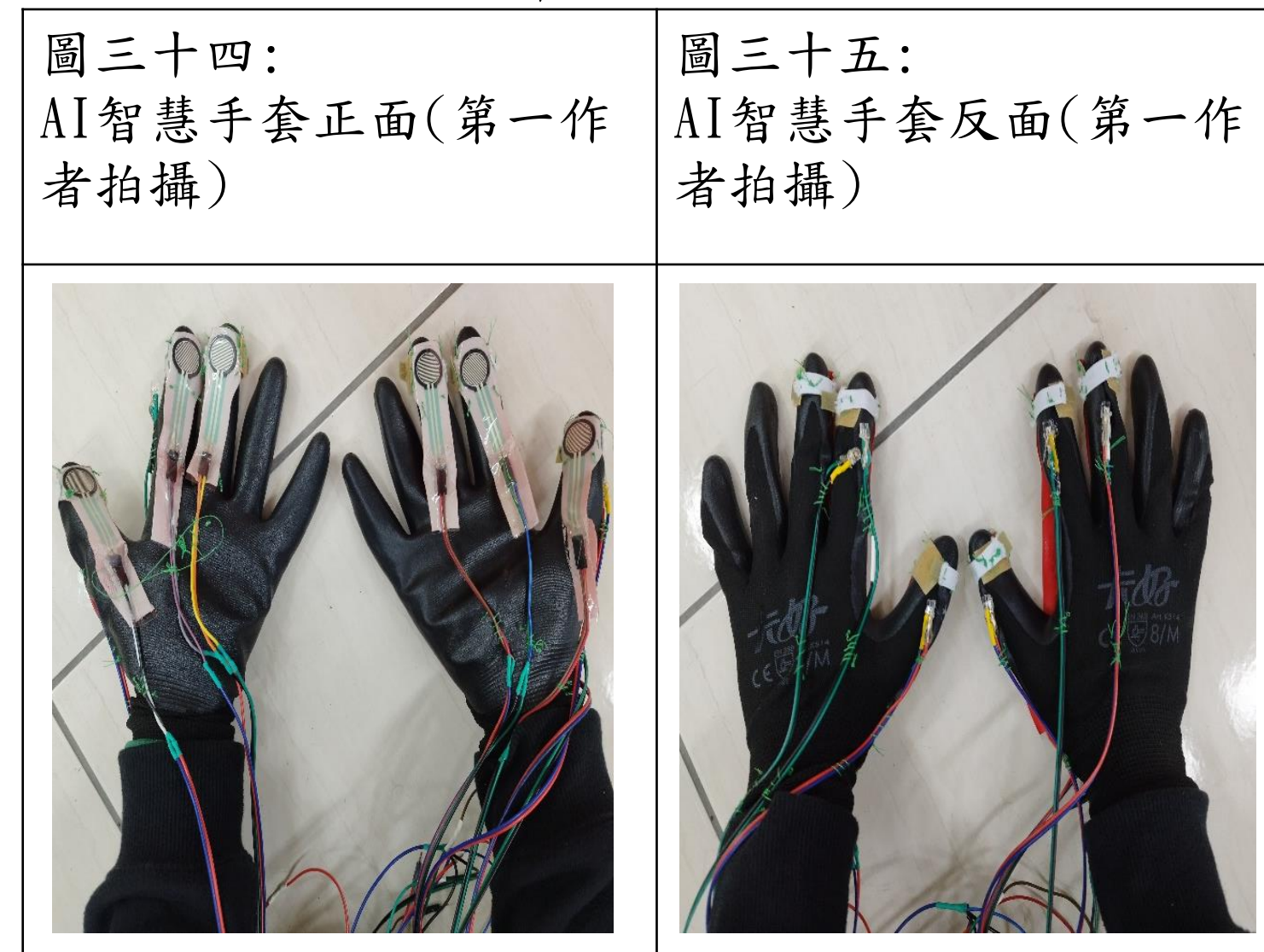
表二: 托球初學者與標準版的關節彎曲角度差

	角A	角B	角C
側面托球標準版	141	179	155
側面托球初學者	106	180	174

十一、測量托球六個手指的碰球時間

(一)在手套上縫製壓力感測器及LED燈泡，如圖三十四、三十五。

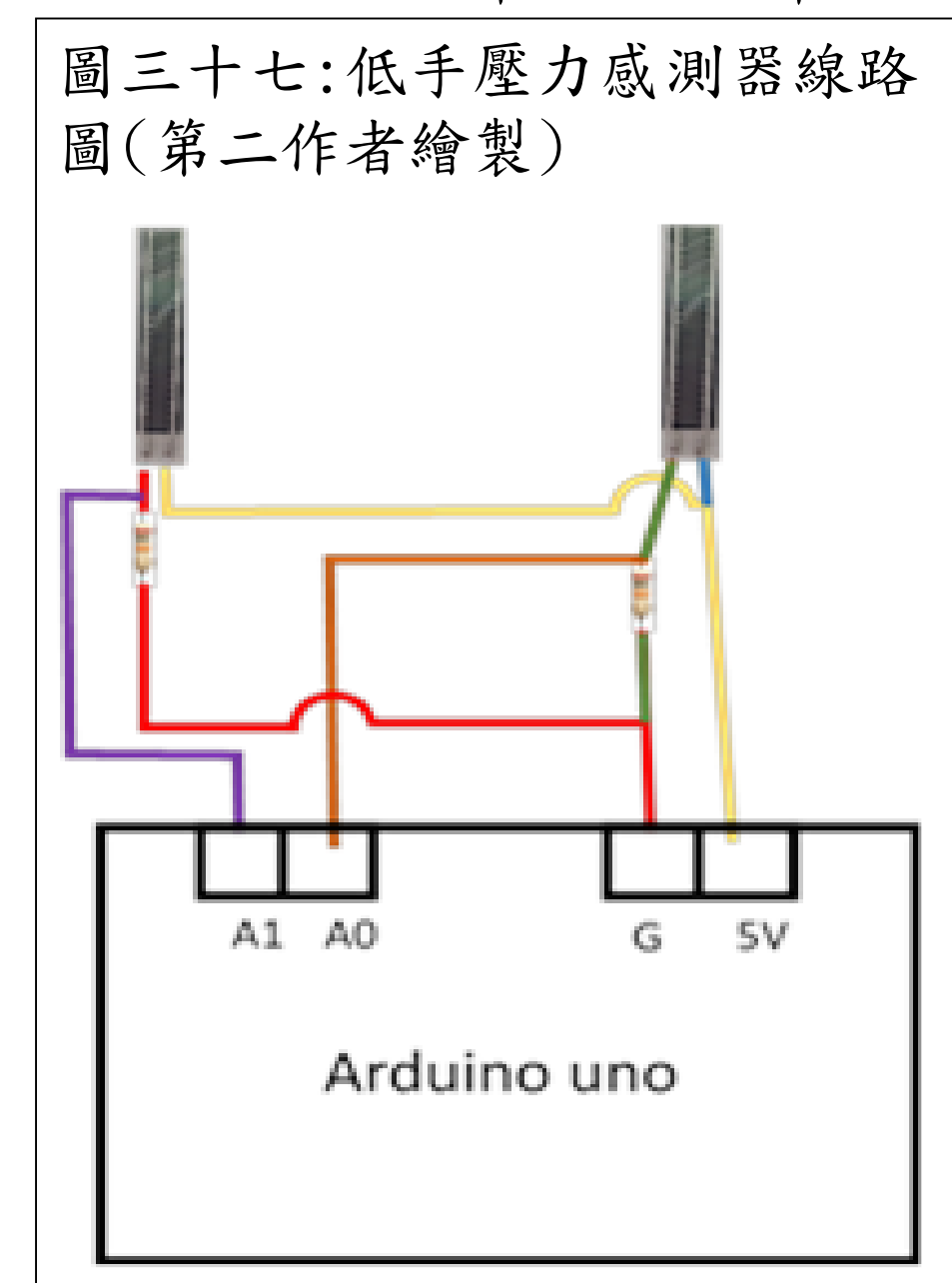
(二)電路圖，如圖三十六



(三)程式碼(詳見附錄七)

十二、製作低手傳球感測袖套

(一)把壓力感測器一端連接5v、一端連接電阻及GND，然後再連接到A0作為右手感測器。左手則連接到A1，作為左手的感測器。如圖三十七、三十八。



(二)程式碼(詳見附錄八)

十三、利用前測與後測比較研究的成果

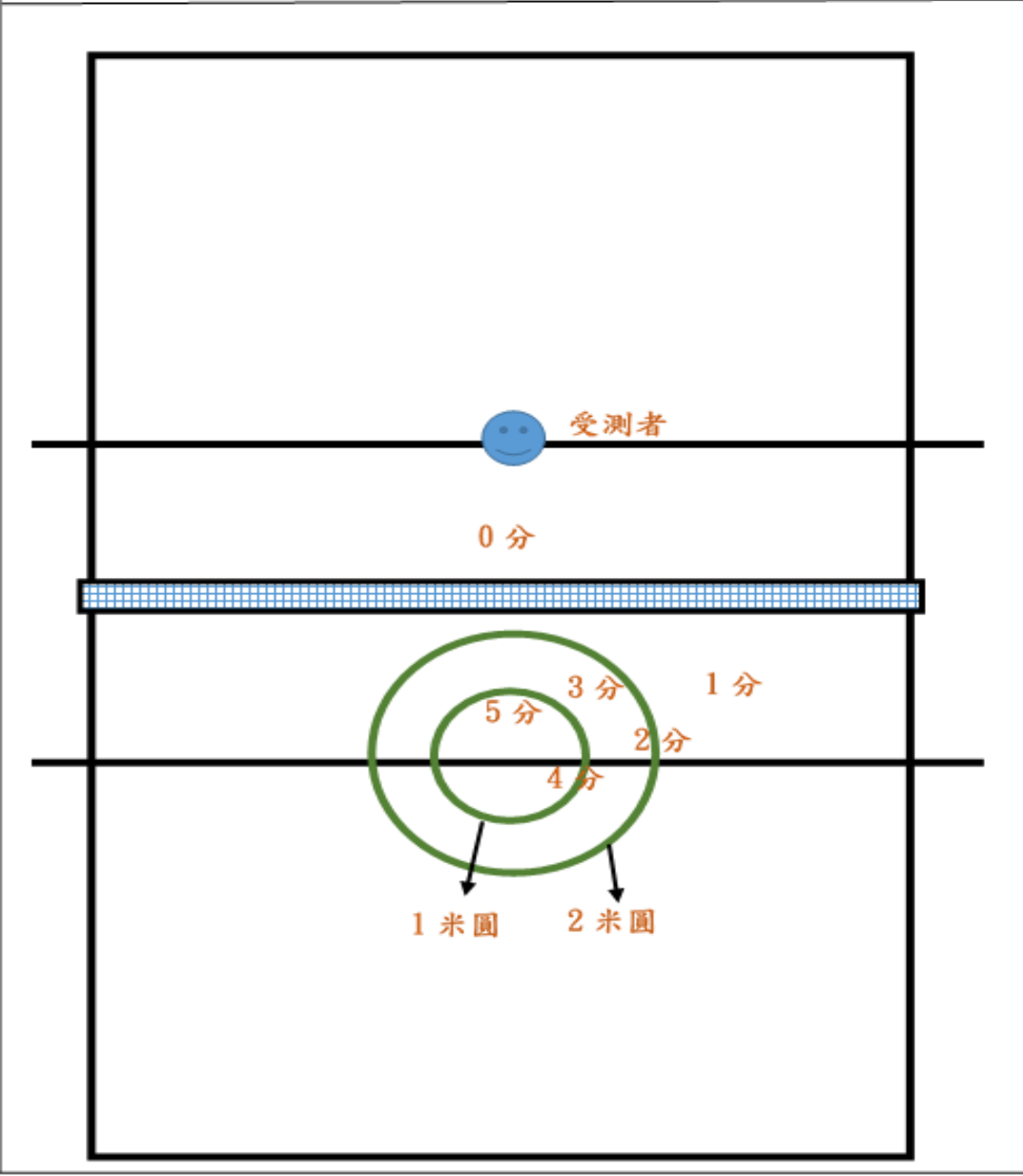
(一)前測:

1. 找出每班同學三男三女，其排球程度分別是好的、中的跟比較差的
2. 每位同學打十球，再根據球的落點計算出總分，如表三、圖三十九

表三：前測計分方式

分數	位置
0分	未過網
1分	接球者在過網界外或未進圈
2分	接球者跨越2米圓
3分	接球者在1米圓到2米圓之間
4分	接球者跨越1米線
5分	接球者在1米圓內或一腳踩在圓上

如圖三十九:測試規則示意圖(第二作者繪製)



3. 前測成績，如表四、表五。

表四：男生前測成績

受測者(男)	A	B	C	D	E
托球	15分	38分	50分	38分	46分
低手	8分	17分	28分	27分	36分

表五：女生前測成績

受測者(女)	A	B	C	D	E
托球	9分	34分	10分	7分	17分
低手	3分	12分	13分	3分	19分

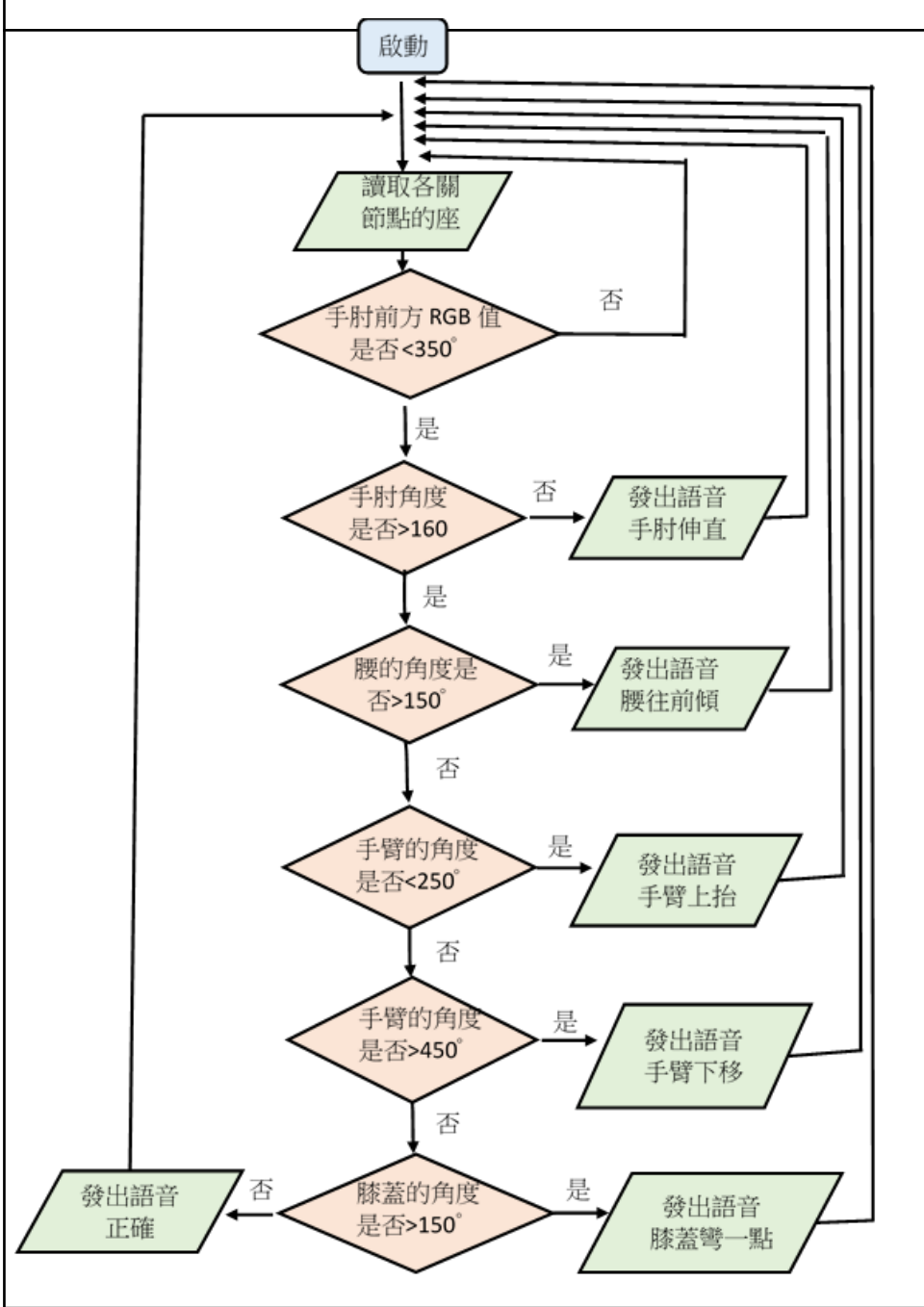
(二)後測:

1. 低手接球

(1) 畫出AI教練流程圖

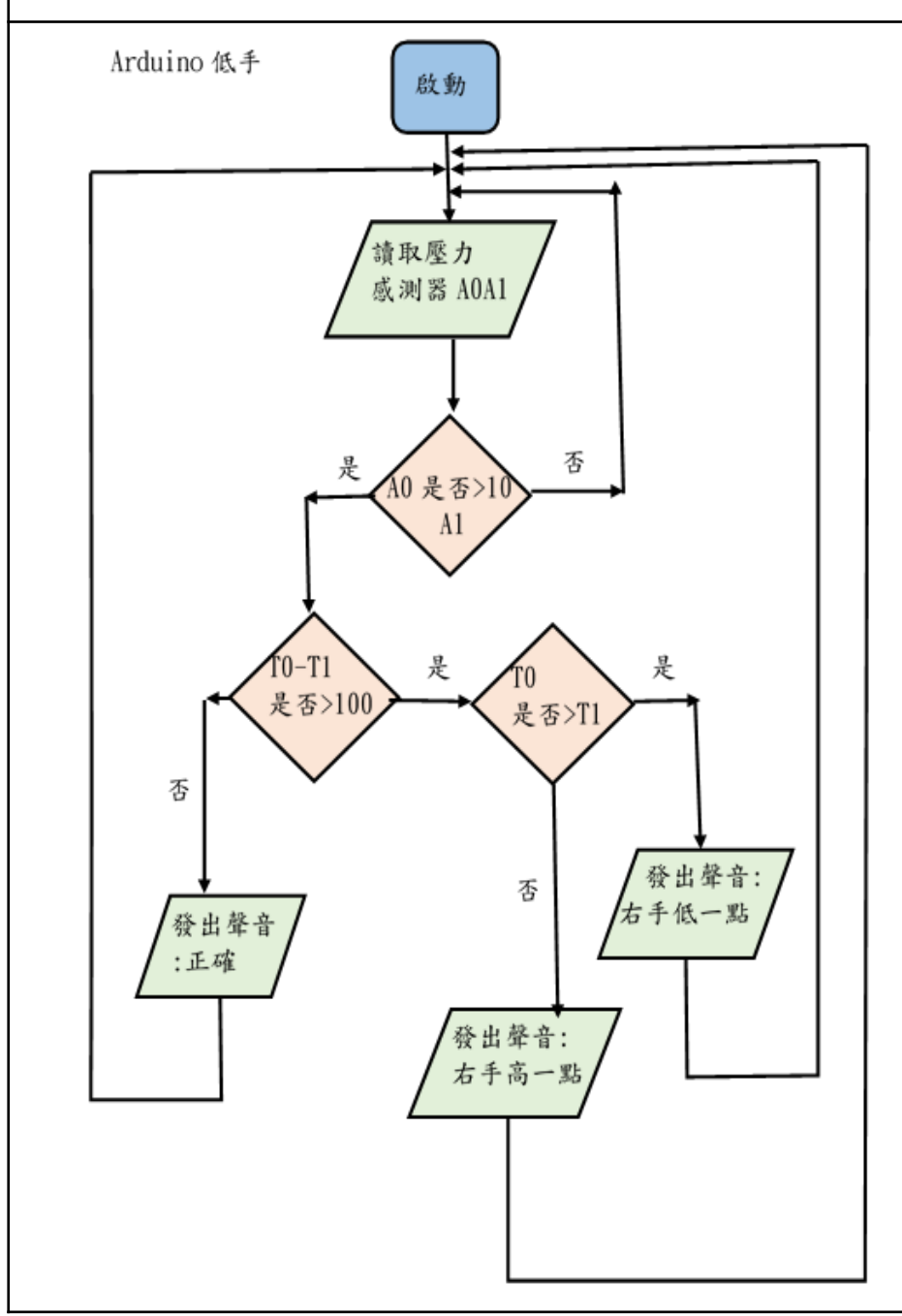
ㄅ. python流程圖

Python低手接球流程示意圖(第二作者繪製)



ㄆ. Arduino流程圖

Arduino低手接球流程示意圖(第二作者繪製)

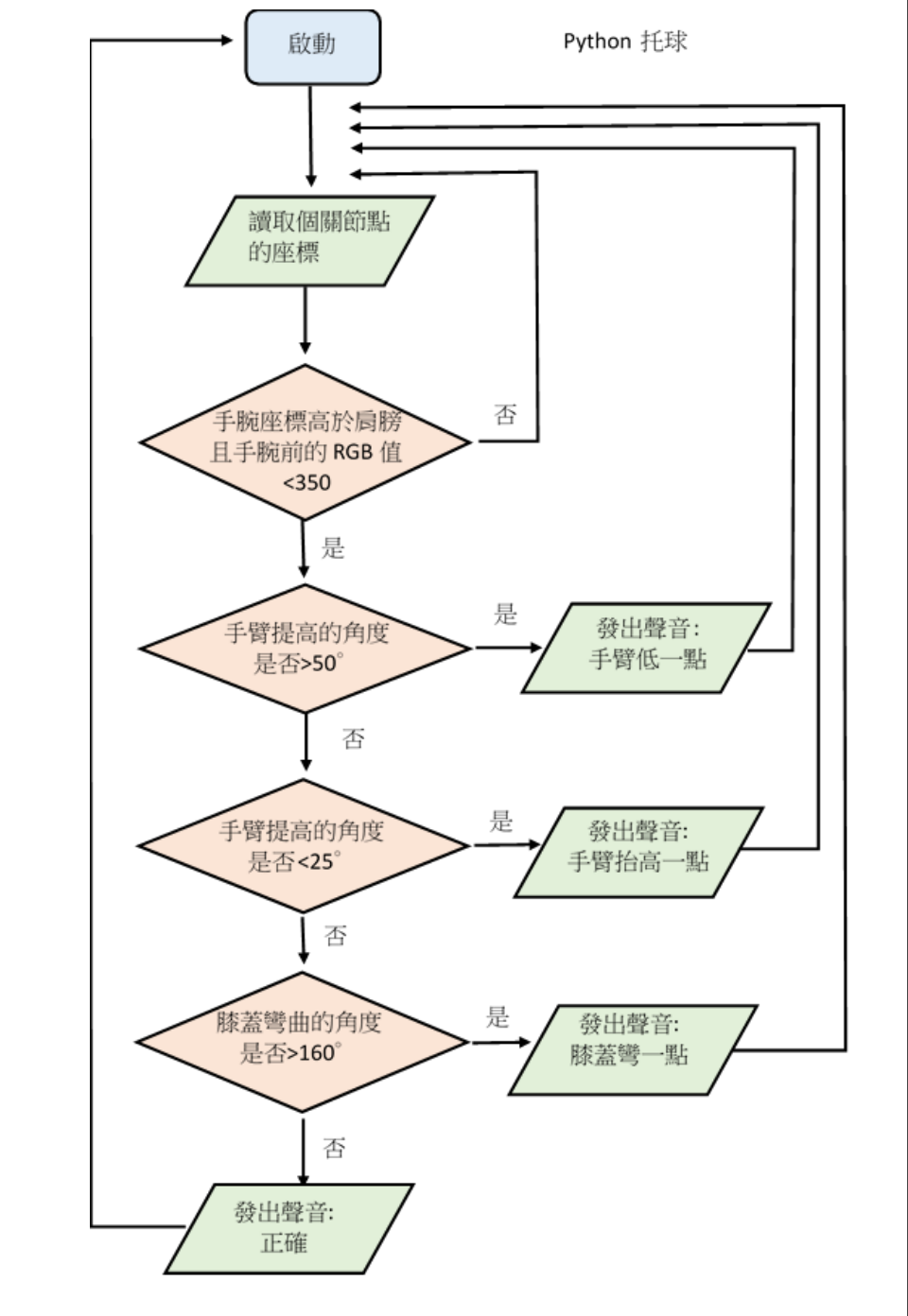


2. 托球

(1) 畫出AI教練流程圖

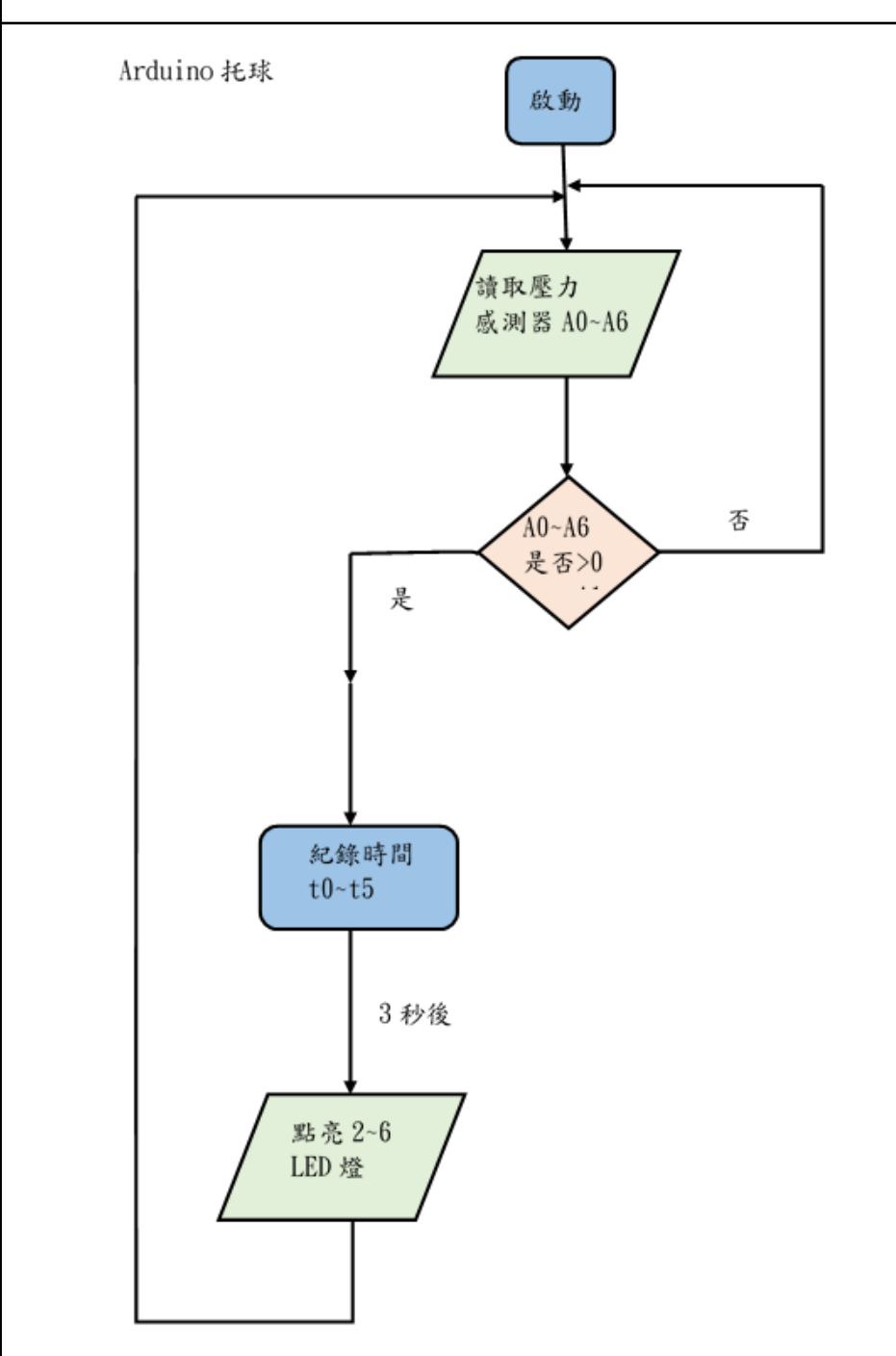
ㄅ. python流程圖

python托球流程示意圖(第三作者繪製)



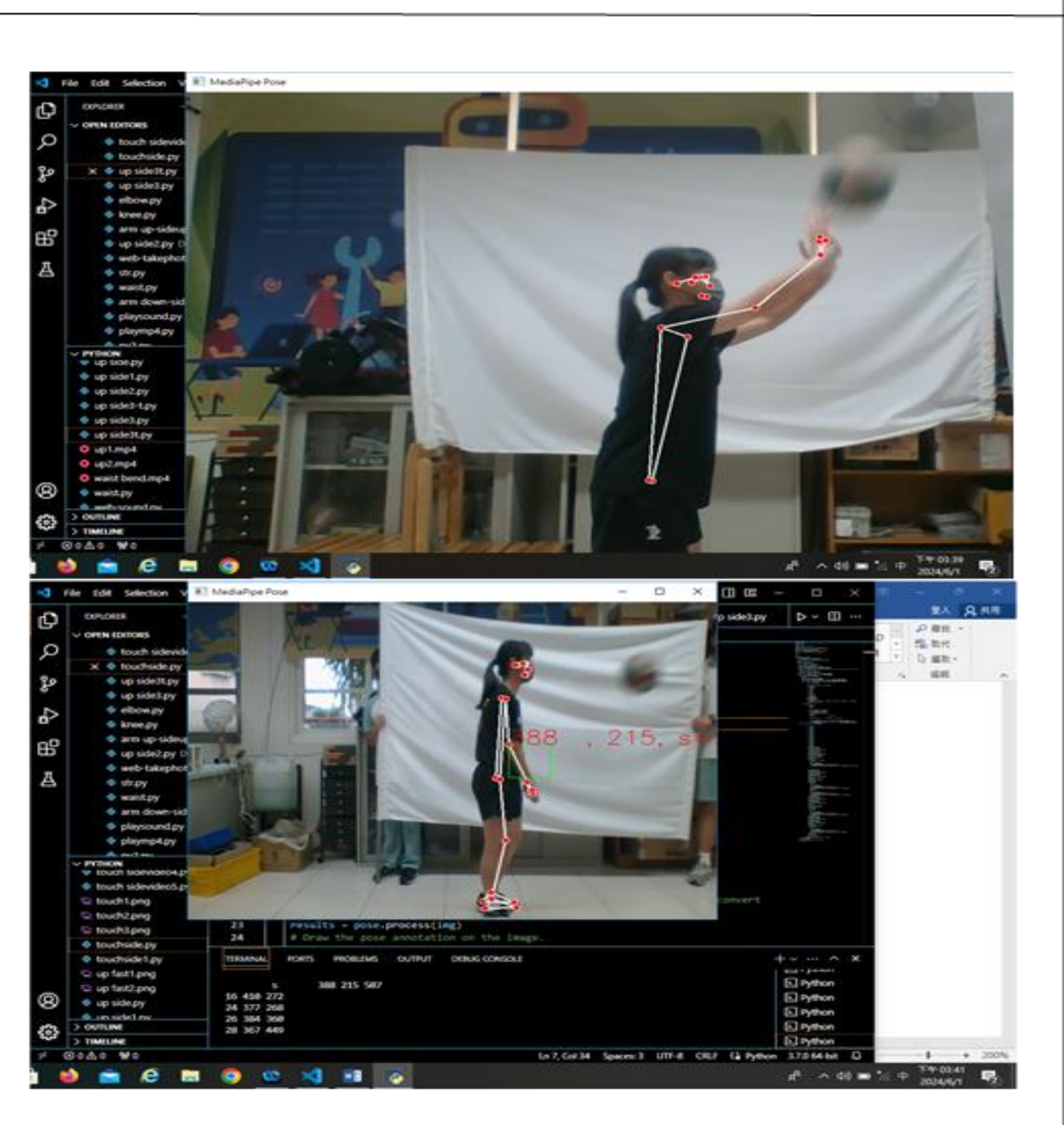
ㄆ. Arduino流程圖

Arduino托球流程示意圖(第三作者繪製)



(2)後測照片如圖四十。

圖四十:檢測接球各關節點角度的電腦畫面(第二作者拍攝)



(3)Python後測成績如表六、表七。

表六：男生python後測成績

受測者(男)	A	B	C	D	E
托球	20分	42分	50分	43分	49分
低手	9分	23分	35分	34分	38分

表七：女生python後測成績

受測者(女)	A	B	C	D	E
托球	15分	41分	15分	33分	45分
低手	9分	24分	27分	25分	38分

(4)戴上穿戴式裝置進行第二次後測成績，如表八、表九。

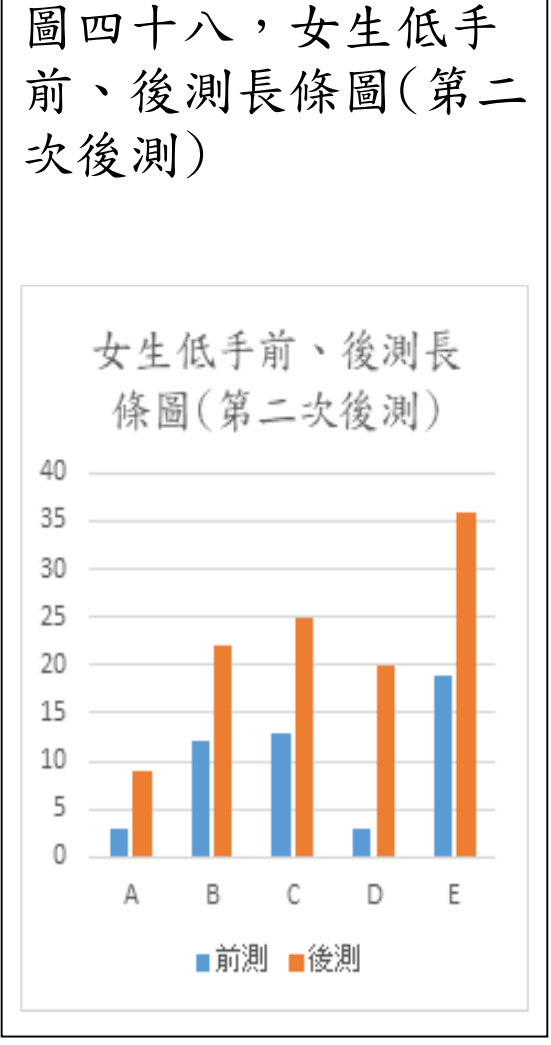
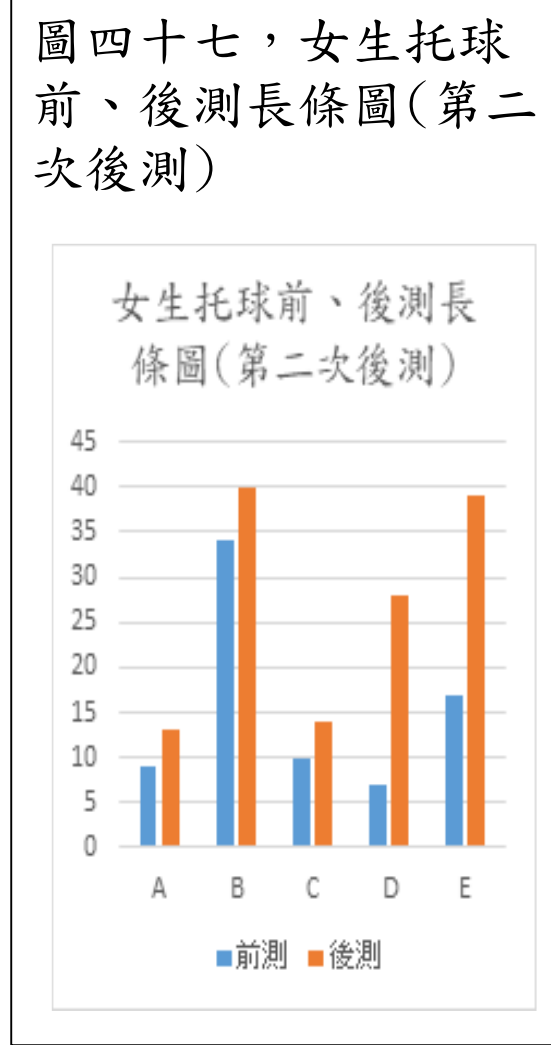
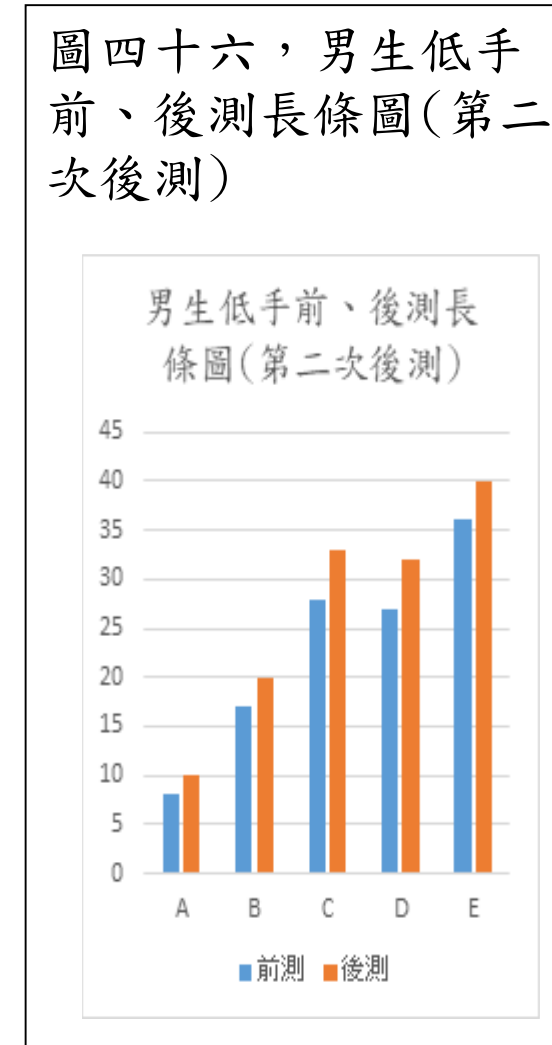
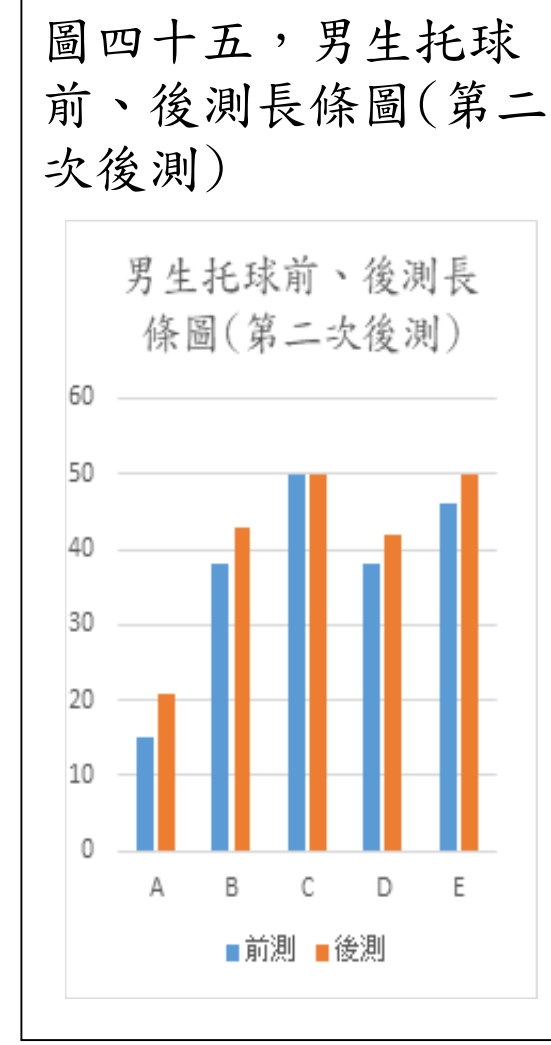
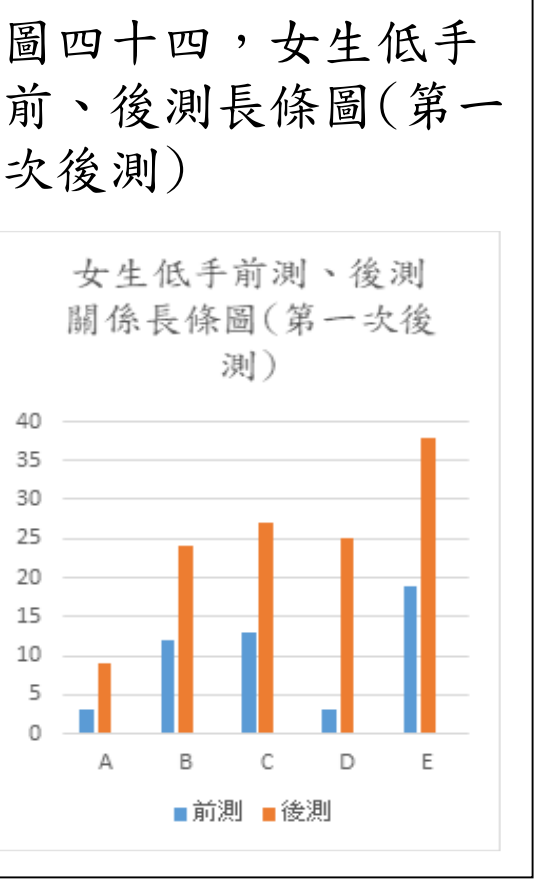
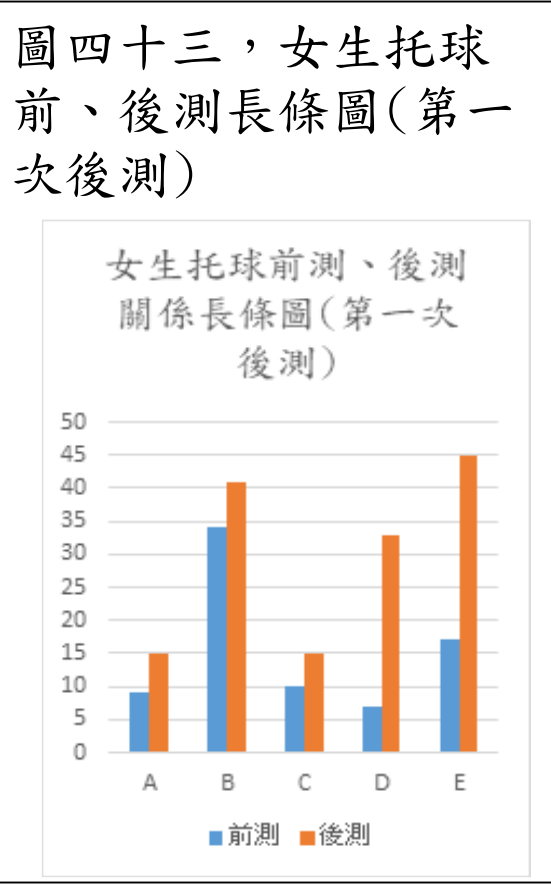
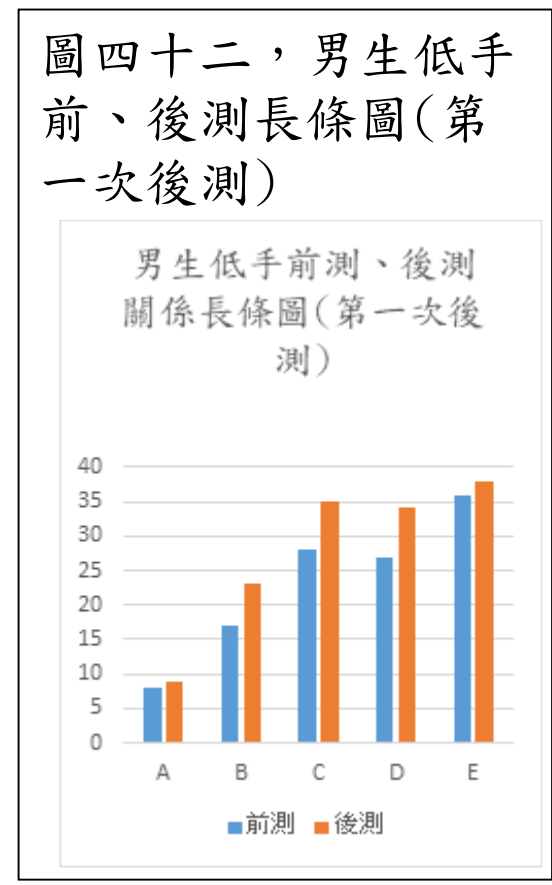
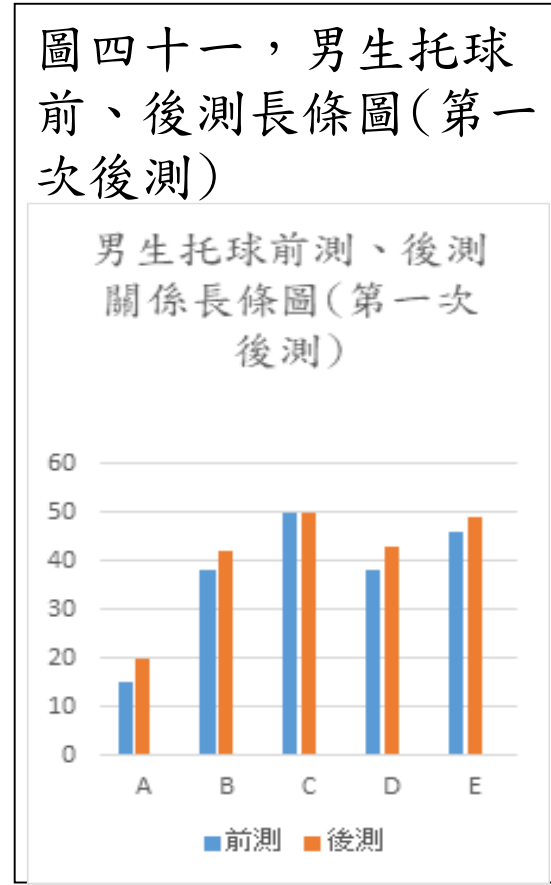
表八：男生穿戴式裝置後測成績

受測者(男)	A	B	C	D	E
托球	21分	43分	50分	42分	50分
低手	10分	20分	33分	32分	40分

表九：女生穿戴式裝置後測成績

受測者(女)	A	B	C	D	E
托球	13分	40分	14分	28分	39分
低手	9分	22分	25分	20分	36分

(5)前測與後測做比較



伍、討論

- 一、在做影像辨識時，拍攝者最好穿緊身的球衣，以免衣服太寬鬆，影響關節位置的判斷。
- 二、用mediapipe來分析動作非常方便，不必再做機器的訓練模式，只要寫好程式即可上手，只要有國中數學的程度，就可以完成。
- 三、本次研究用的軟體如python、opencv、mediapipe、arduino，都是免費的，可以在經費有限的情況下，做最大的發揮。
- 四、本研究針對的是基礎的排球動作，接球都是正面回擊到正前方，沒有傳球到特定角度，如此處理起來較為單純。
- 五、每位排球選手在擊球前的準備動作都不盡相同，但是擊球的瞬間動作一致，所以分析動作時，只擷取擊球瞬間的照片即可。
- 六、由於打排球的時候，人的移動不會太快，所以用一般的攝影機就可以，不會有殘影出現。除非是要分析扣球的動作，才用到高速攝影。所以對於基礎運動員來說，一般的設備就足夠了

陸、結論

- 一、mediapipe軟體是由 Google Research 所開發的多媒體機器學習模型，該公司已經針對不同種族、年齡的人檢測，可以充分辨識人的姿勢、表情，所以藉由mediapipe我們可以輕鬆地跨入影像辨識的領域。
- 二、計算角度需用到三角函數，而程式碼算出的角度是弧度，也就一個圓的角度是 2π ，與我們熟悉的一個圓360度不一樣，所以要必須經過數學換算，才能變成我們熟悉的角度。
- 三、在受測者的運動服不太寬鬆的情況下，AI排球教練均能辨識出受測者各個關節的角度，誤差不會太大，所以用AI教練來判斷受測者的動作是否標準可行。但是對於細部的托球動作，失之毫釐，差之千里，無法用影像辨識的方法來判斷手指的動作，所以用壓力感測器配合 ARDUINO來顯示各個手指觸球的時間，可以精確的判斷各個手指的觸球時間差距，作為運動員修正的參考。
- 四、在科技日新月異的時代，運動員的訓練方式已經科技化，但排球運動的訓練模式仍是一層不變，都是由教練用眼睛看，再指導學生修正錯誤。這種方法沿用了數十年，教練一次只能指導一個學生，效率較低，學生也因為看不到自己的動作與標準動作差哪裡，修正起來與教練的要求有落差。藉由AI教練，運動員可以有客製化的學習方式，可以看到自己的缺點，也可以了解到修正的方式，效果強，值得來推廣。
- 五、未來AI排球教練還可以分析擊球者的表情，判斷是要舉出哪一種戰術，也可以判斷攻擊手的動作或表情，是真正要攻擊還是虛晃一招，藉由這些功能的延伸，可以提升排球的水準，讓AI排球教練由動作矯正擴展到戰術判斷，變成功能全面化的AI教練。
- 六、經過前測與後測的比較，發現大部分的選手在後測後成績都有顯著的提升，可見及時矯正受測者的動作對於受測者學習排球的成效來說是有幫助的。所以利用AI排球教練的軟體可以有效的提升受測者學習排球的效果，可以節省人力，是一種非常方便的輔助教學器材。

柒、參考文獻資料

- 一、STEAM教育學習網。
- 二、鄧文淵，文淵閣工作室，Python初學特訓班(第三版):從快速入門到主流應用 全面實戰，2019年12月三版，基峰資訊股份有限公司
- 三、Martin Evans, Joshua Noble, Jordan Hochenbaum作，王冠勳譯，Arduino完全實戰手冊，2014年2月，博碩文化股份有限公司
- 四、楊明豐，Arduino最佳入門與應用-----打造互動設計輕鬆學2014年5月初版二刷，基峰資訊有限公司