

人工智能射箭姿勢檢測系統



Artemis_AI_Archery | KSLO_STEAM_CLUB



李奕潼老師

- Artemis AI 射箭姿勢檢測系統 指導老師
- STEAM學會負責老師
- STEAM Task Force成員
- 學校箭藝隊領隊
- 香港射箭總會認可教練
- 現為香港射箭集訓隊成員





Artemis 人工智能射箭姿勢檢測系統

- 由學校STEAM CLUB 幹事發起的STEAM運動專案
- EDU STEAMlympics 2024 中學組冠軍
- Maker Faire Shenzhen 2024 最佳展商
- Maker Faire Taipei 2024 參展商



學生開始專案的契機



每次STEAM比賽都被嘗新想法的出現難倒
在空檔期突發奇想，想到「用科技使運動更加科學和系統化」作下一次比賽的提案



- 剛好遇上EDU STEAMlympics 比賽
- 結合學校需要及指導老師的興趣範疇
- 決定以箭藝一運動作實踐目標
- 「AI 射箭姿勢檢測系統」就此誕生



專案背景：需要

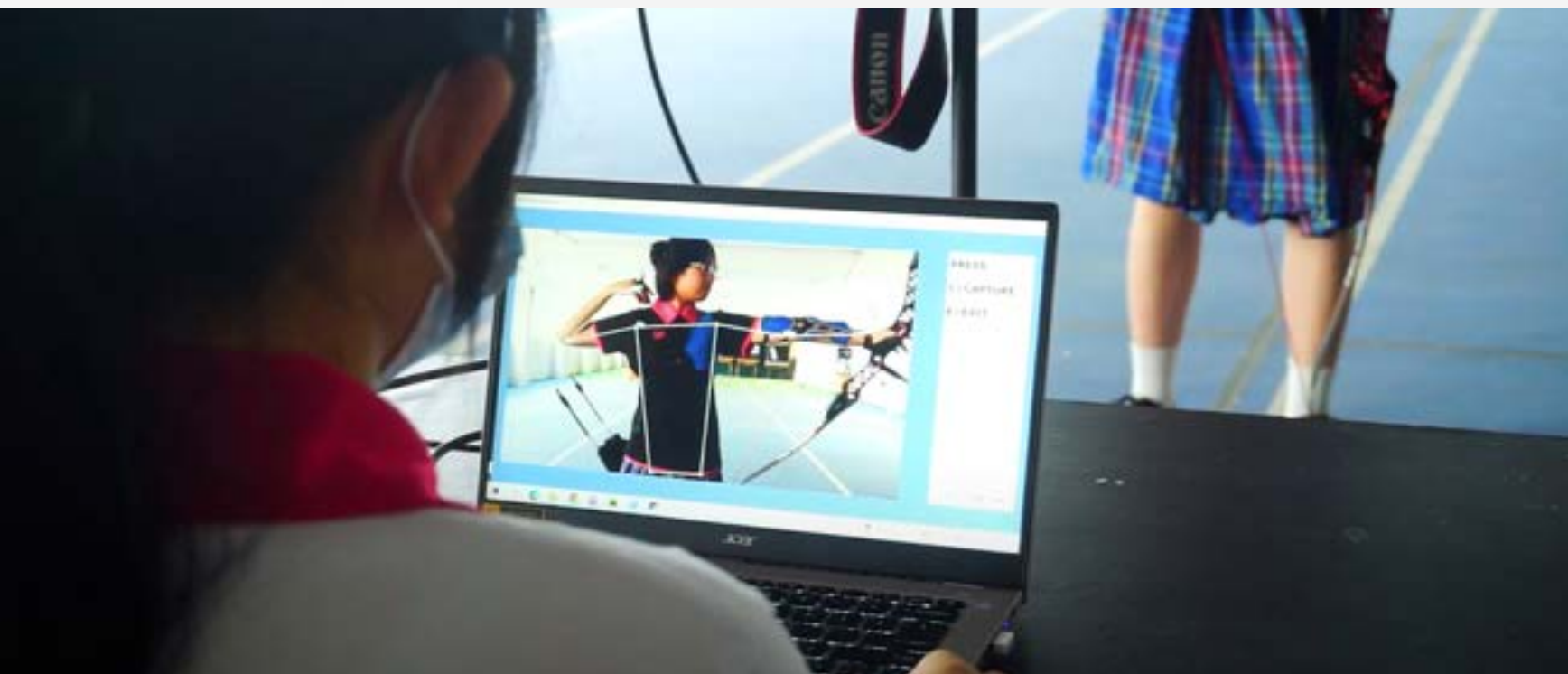
- 射箭隊是本校重點發展活動，於射箭界別亦頗有知名度，為了成為更有競爭力的隊伍，領隊老師開始思考「如何令練習更加科學」。
- 隊伍人數大幅上升，造成人手不足，單靠人力難以照顧。
- 此外隨隊員技術提升，肉眼難辨的微小動作差距需精準工具判別。借此希望單靠感覺的動作能夠可視化，可量度地呈現在學生面前。
- 帶動學生自主檢視練習成果，於自主練習時維持質素。



專案背景：目標

1. 縮短教練對每位隊員的指導時間
2. 令訓練更科學及系統化
3. 可視化、可量度的技術指導，而非單靠感覺的抽象論述
4. 學生用作自主練習或檢討的工具
5. 記錄學生姿勢動作





Artemis 人工智能射箭姿勢檢測系統

射箭是講求一致性的運動，需要穩定的動作框架，每個重點關節點的角度是重要因素。Artemis 希望透過AI Archery Pose達至可視化、可量度的技術指導，而非單靠感覺的抽象論述，甚至成為學生的自主練習、檢討的工具。

主要功能：

- 關節角度檢視功能
- 畫面延遲功能
- 自動截圖功能

射箭運動介紹

- 射箭是講求一致性的運動，需要穩定的動作框架
- 每個關節點的角度是其中一個重要因素
- 每個人的骨架不同，動作需要量身定制，有一定複雜性



射箭運動介紹

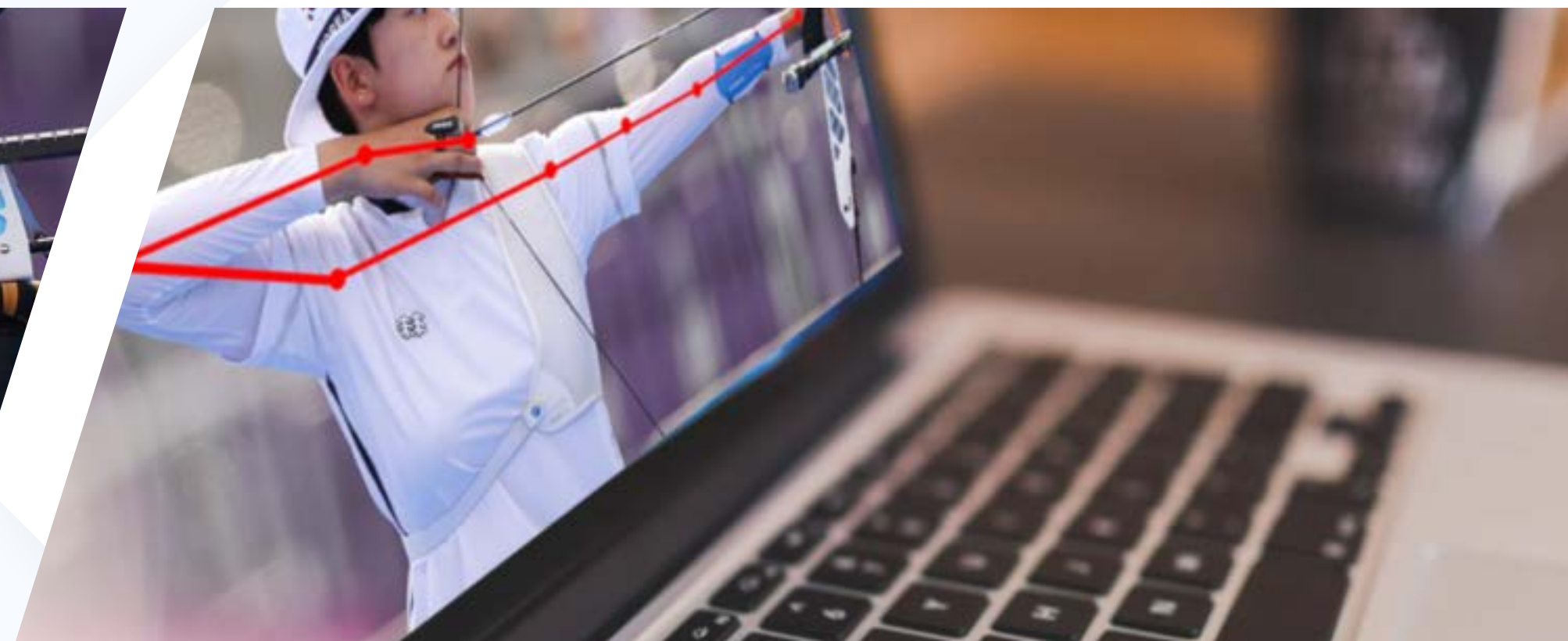
重點關節點

- 推弓點
- 肩膊
- 壓弦定位
- 手腕
- 手肘

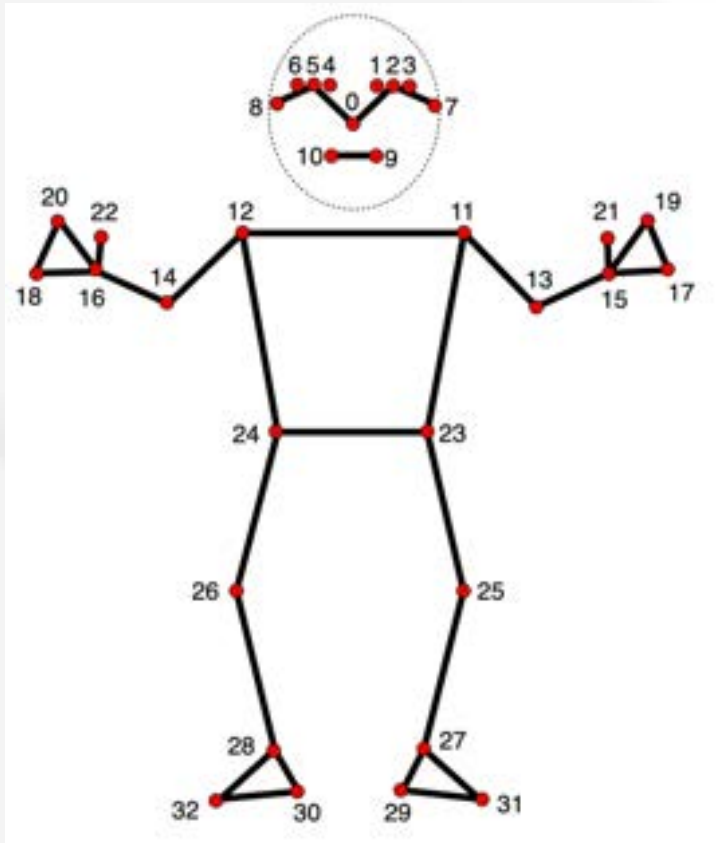


專案構想

- 利用AI 辨別姿勢節點位置，做出能檢測射箭姿勢的程式
- 設備: 電腦，鏡頭，腳架
- 平台: Python
- Framework: Mediapipe



```
PoseLandmarkerResult:
  Landmarks:
    Landmark #0:
      x      : 0.638852
      y      : 0.671197
      z      : 0.129959
      visibility : 0.9999997615814209
      presence  : 0.9999984502792358
    Landmark #1:
      x      : 0.634599
      y      : 0.536441
      z      : -0.06984
      visibility : 0.999909
      presence  : 0.999958
    ... (33 landmarks per pose)
  WorldLandmarks:
    Landmark #0:
      x      : 0.067485
      y      : 0.03108
      z      : 0.05522
      visibility : 0.99999
      presence  : 0.99999
    Landmark #1:
      x      : 0.06320
      y      : -0.0038
      z      : 0.02092
      visibility : 0.99997
      presence  : 0.99999
    ... (33 world landmarks)
  SegmentationMasks:
    ... (pictured below)
```



Mediapipe (Framework)

- 在應用程式中快速套用人工智慧 (AI) 和機器學習 (ML) 技術，並依據需求進行自訂，在多個開發平台上使用
- 免費的開源框架

如何用MEDIAPIPE 切入射箭動作?



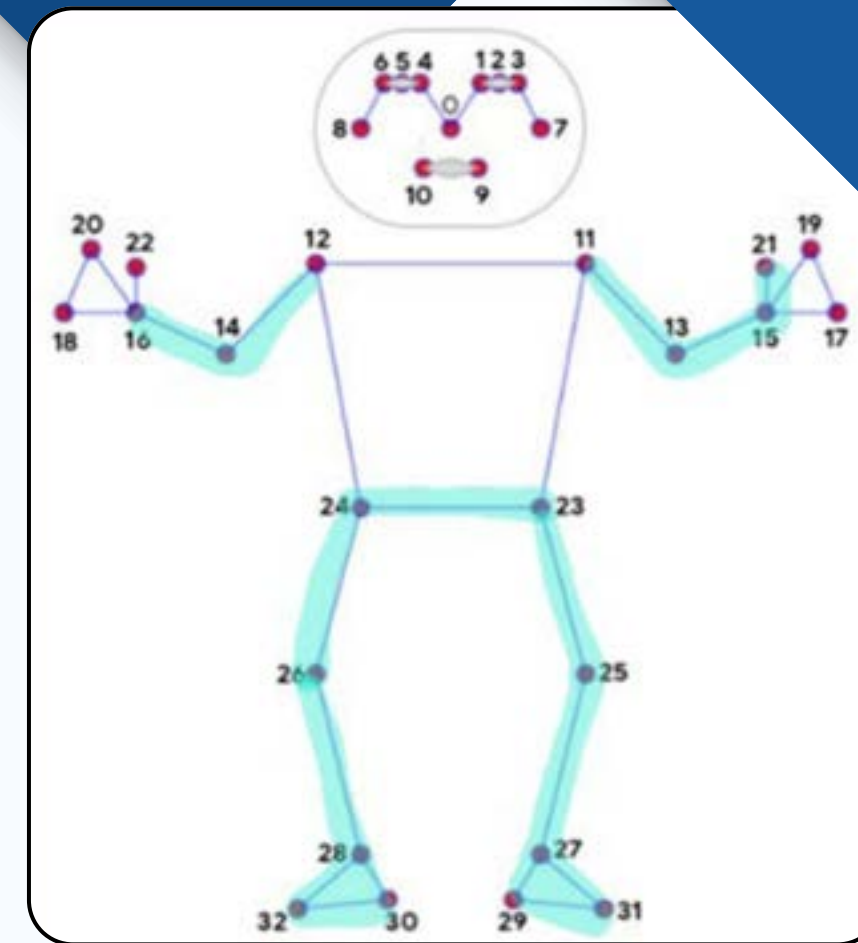
姿勢偵測:

- 透過攝影鏡頭獲取影像後，即時標記身體骨架和動作
- 透過程式語言演算計算角度，找出一致性程度



作用:

- 作出更具體的練習指示
- 幫助學生了解自身動作→改善射箭的姿勢
- 可使用手部動作檢測分析放箭情況
- 教練能有可視化、可量度的技術指導



重點關節點的對應節點:

11-16, 21(推弓點、肩膊、壓弦定位、手腕、手肘)

```
# 啟用姿勢偵測
with mp_pose.Pose(
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as pose:
```

```
results = pose.process(img2)
# 取得姿勢偵測結果
# 根據姿勢偵測結果，標記身體節點和
# 骨架
mp_drawing.draw_landmarks(
    img,
    results.pose_landmarks,
    mp_pose.POSE_CONNECTION
```

本專案如何切合本校及射手需要？

困難：

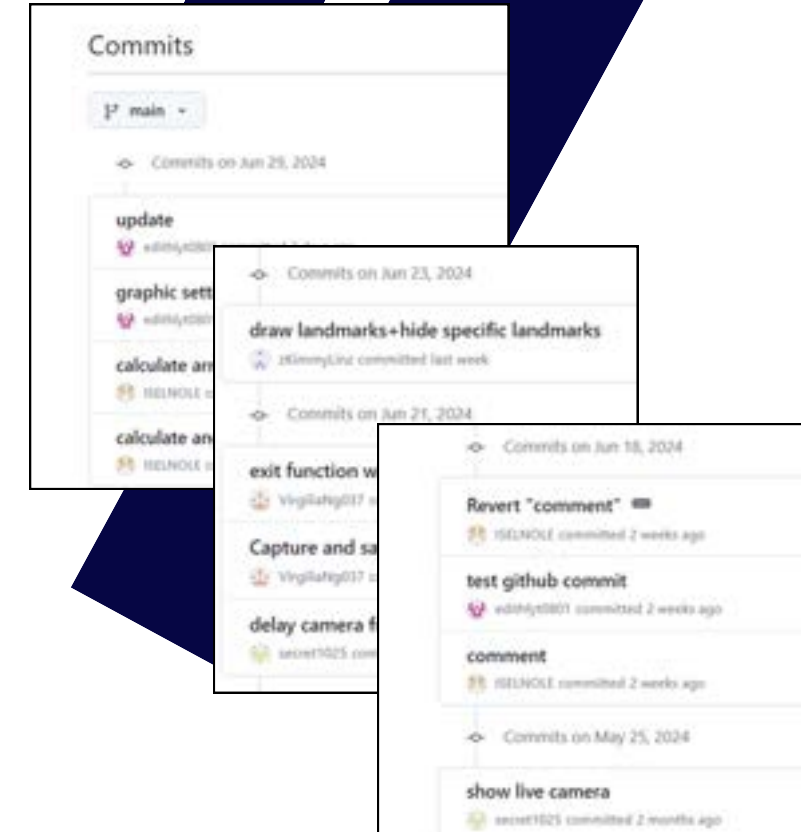
- 射手人數>教練人數
- 教練無法時刻監管射手姿勢
- 部分射手難以發現自己出現錯誤姿勢
- 人眼無法精準判斷姿勢對錯

專案可以：

- 減輕教練及輔導員負擔
- 射手能夠自主練習
- 精確改善射手的不足
- 提高訓練效率
- 令細小的錯誤可視化
- 提升箭藝隊水平



TIMELINE



3—4月

討論及擬定專案框架
相關資料搜集
初步分工
預備簡報及線上演講

4—5月

帳號開設
軟件下載及電腦設定
網上資源自習
(Python, Github)

5—6月

程式編寫
程式測試

6—7月

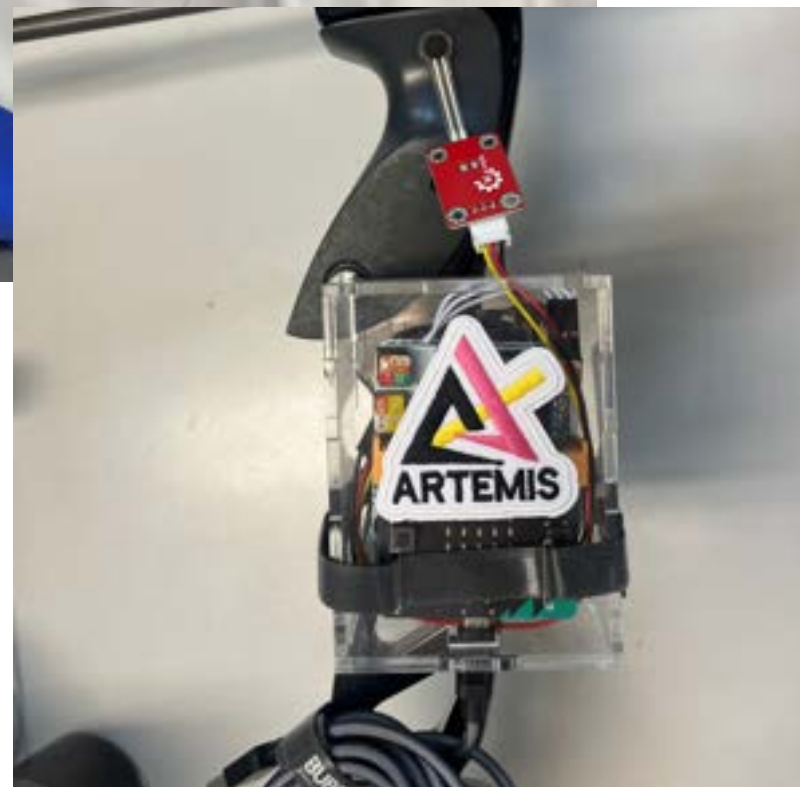
作品展示視頻拍攝
作品展示視頻剪輯

TIMELINE

9-10月

作品優化

EDU STEAMlympics 展出及決賽



TIMELINE

10月至今

帶著Artemis參與大大小小的活動



Instructor's Timeline



3—4月

- 協助討論及擬定專案框架
- 引導同學搜集相關資料
- 按學生長處分工
- 安排任務及時間表
- 指導簡報製作及線上演講



4—5月

- 協助學生開設帳號
- 協助軟件下載及電腦設定
- 搜集及篩選網上自習資源
- (Python, Github)



5—6月

- 引導同學程式編寫
- 協助進行程式測試
- 與同學共同進行程式除錯



6—7月

- 指導學生運行視頻拍攝
- 指導作品展示視頻剪輯

7月後就是不斷DEBUG、優化和出席活動....

Instructor's Timeline

3—4月

- 協助討論及擬定專案框架
- 引導同學搜集相關資料
- 按學生長處分工
- 安排任務及時間表
- 指導簡報製作及線上演講

4—5月

- 協助學生開設帳號
- 協助軟件下載及電腦設定
- 搜集及篩選網上自習資源
- (Python, Github)

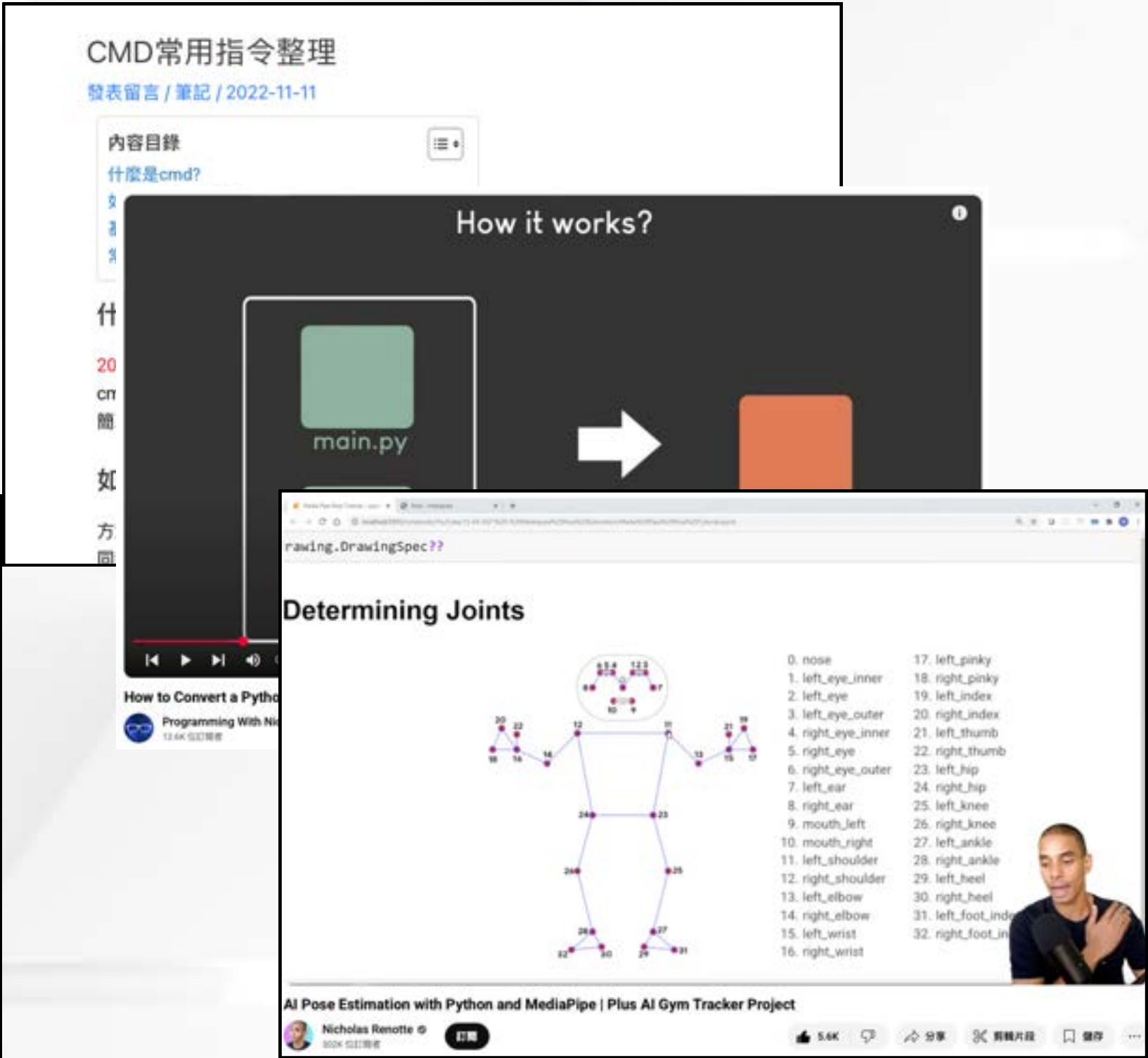
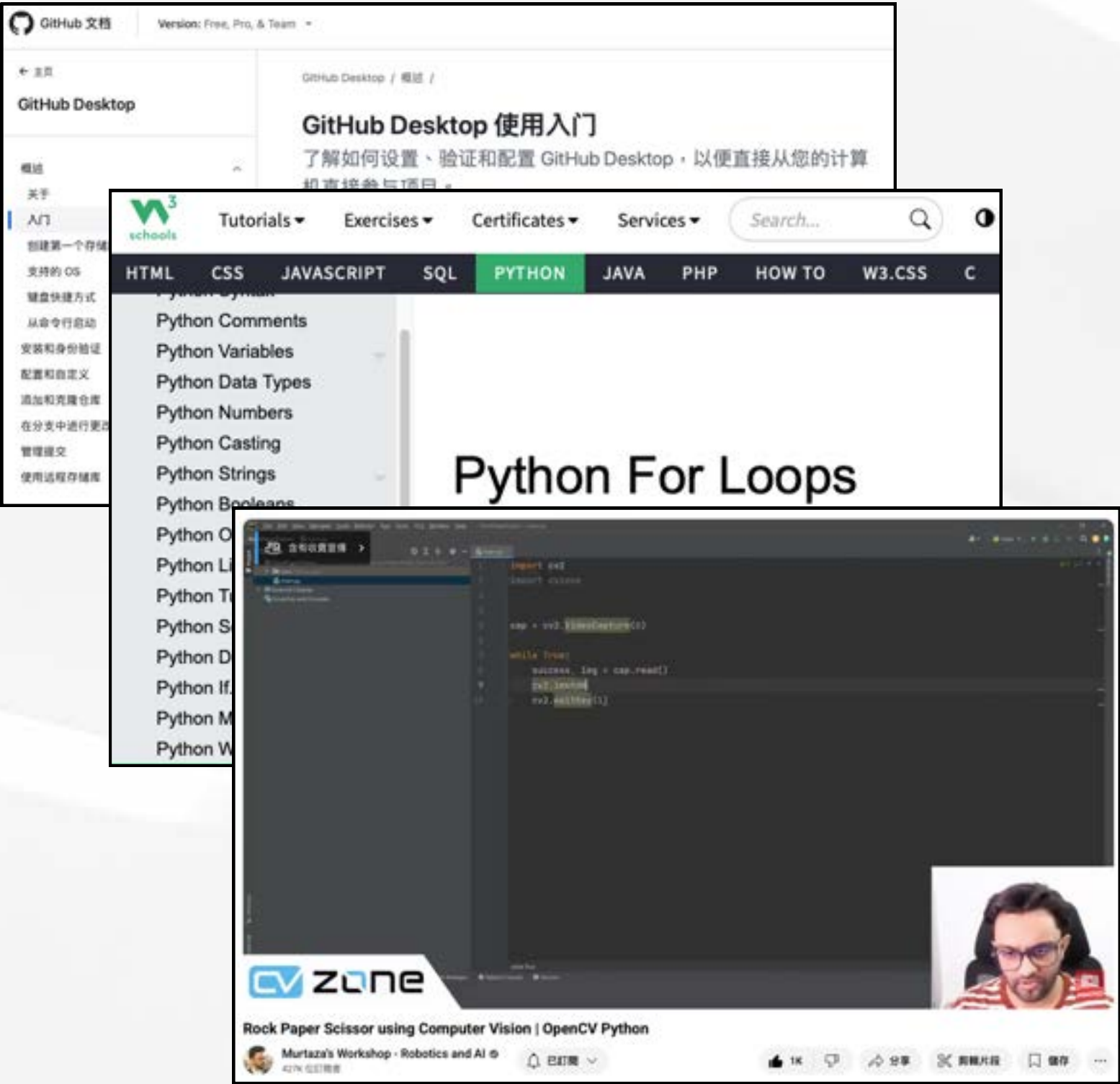
5—6月

- 引導同學程式編寫
- 協助進程式測試
- 與同學共同進程式除錯

6—7月

- 指導學生運行視頻拍攝
- 指導作品展示視頻剪輯

7月後就是不斷DEBUG、優化和出席活動....



Visualization utilities

```
In [ ]: ##markdown To better demonstrate the Pose Landmarker API, we have created a set of visualization tools that will

from mediapipe import solutions
from mediapipe.framework.formats import landmark_pb2
import numpy as np

def draw_landmarks_on_image(rgb_image, detection_result):
    pose_landmarks_list = detection_result.pose_landmarks
    annotated_image = np.copy(rgb_image)

    # Loop through the detected poses to visualize.
    for idx in range(len(pose_landmarks_list)):
        pose_landmarks = pose_landmarks_list[idx]

        # Draw the pose landmarks.
        pose_landmarks_proto = landmark_pb2.NormalizedLandmarkList()
        pose_landmarks_proto.landmark.extend([
            landmark_pb2.NormalizedLandmark(x=landmark.x, y=landmark.y, z=1
        )])
        solutions.drawing_utils.draw_landmarks(
            annotated_image,
            pose_landmarks_proto,
            solutions.pose.POSE_CONNECTIONS,
            solutions.drawing_styles.get_default_pose_landmarks_style())
    return annotated_image
```

```
In [ ]: # STEP 1: Import the necessary modules.
import mediapipe as mp
from mediapipe.tasks import python
from mediapipe.tasks.python import vision

# STEP 2: Create an PoseLandmarker object.
base_options = python.BaseOptions(model_asset_path='pose_landmarker.task')
options = vision.PoseLandmarkerOptions(
    base_options=base_options,
    output_segmentation_masks=True)
detector = vision.PoseLandmarker.create_from_options(options)

# STEP 3: Load the input image.
image = mp.Image.create_from_file("image.jpg")

# STEP 4: Detect pose landmarks from the input image.
detection_result = detector.detect(image)

# STEP 5: Process the detection result. In this case, visualize it.
annotated_image = draw_landmarks_on_image(image.numpy_view(), detection_result)
cv2_imshow(cv2.cvtColor(annotated_image, cv2.COLOR_RGB2BGR))
```

```
def calculate_angle(a, b, c): #https://www.geeksforgeeks.org/numpy-arctan2-python/
    a = np.array(a) # First
    b = np.array(b) # Mid
    c = np.array(c) # End

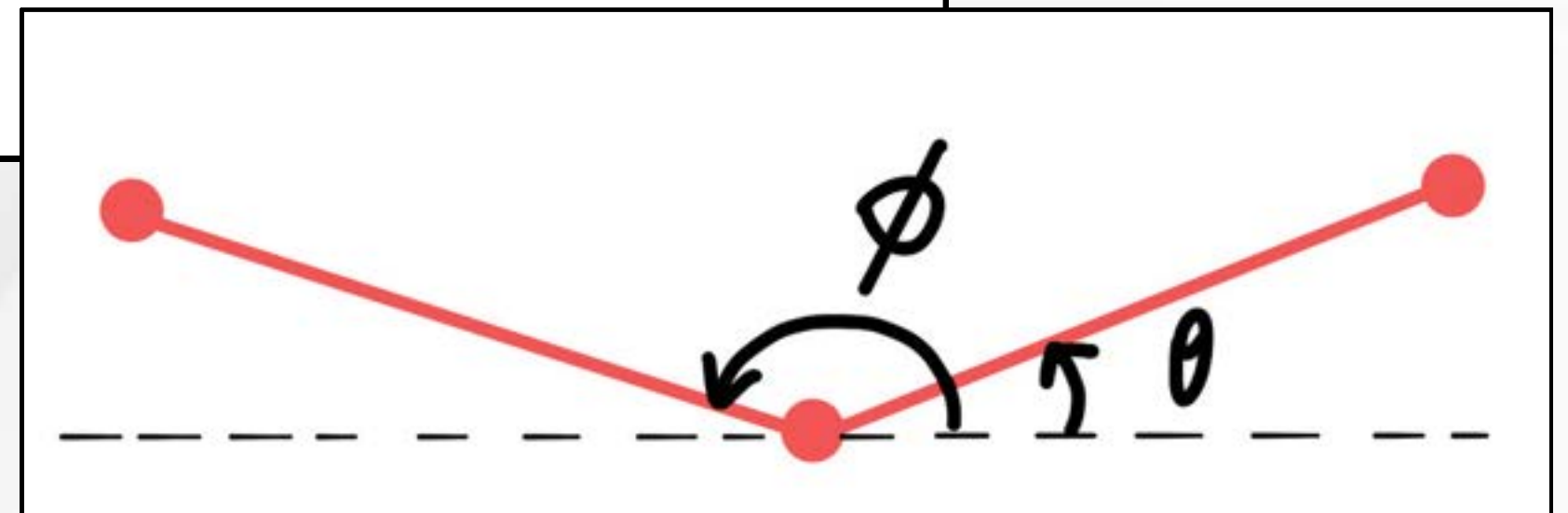
    radians = np.arctan2(c[1] - b[1], c[0] - b[0]) - np.arctan2(a[1] - b[1], a[0] - b[0])
    angle = np.abs(radians * 180.0 / np.pi)

    if angle > 180.0:
        angle = 360 - angle

    return angle
```

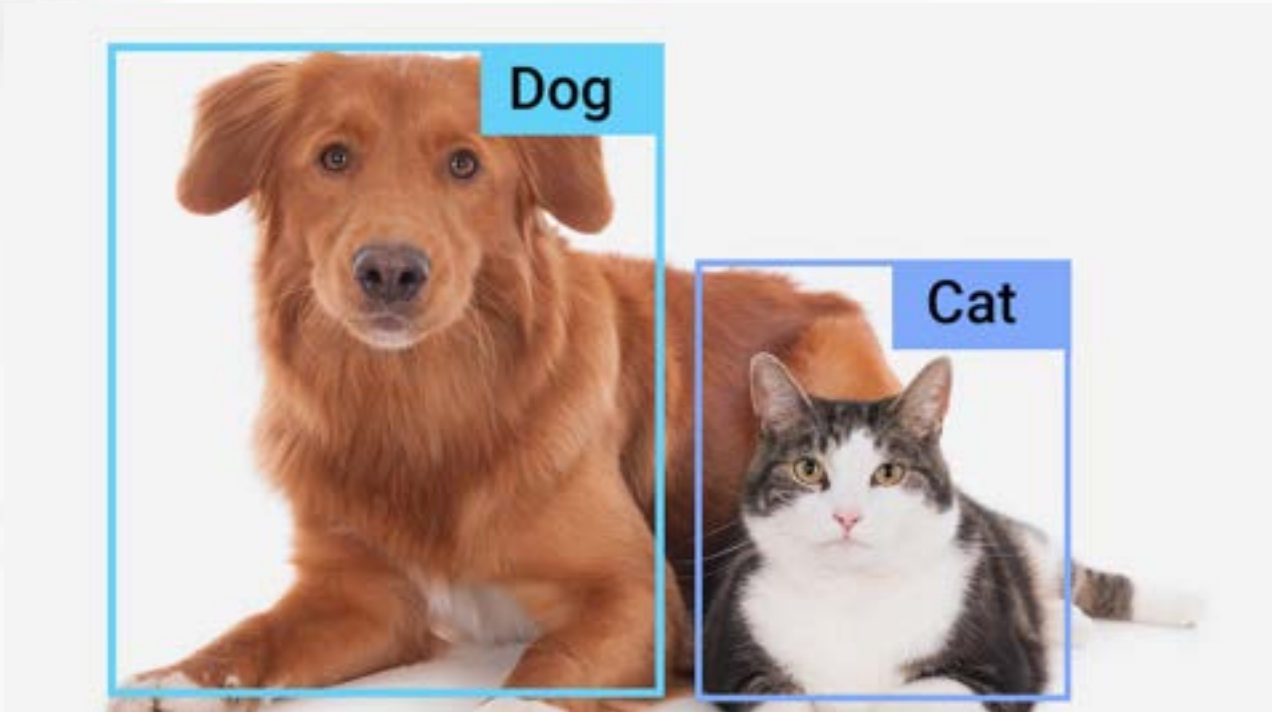
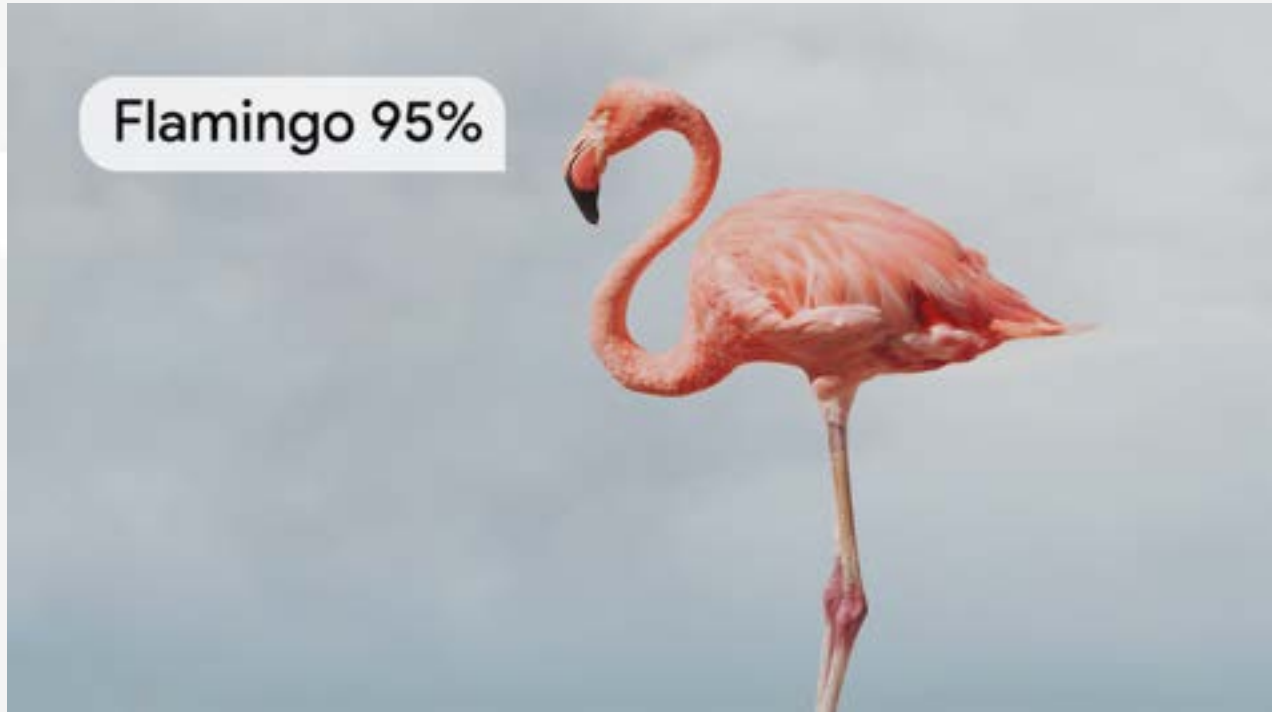
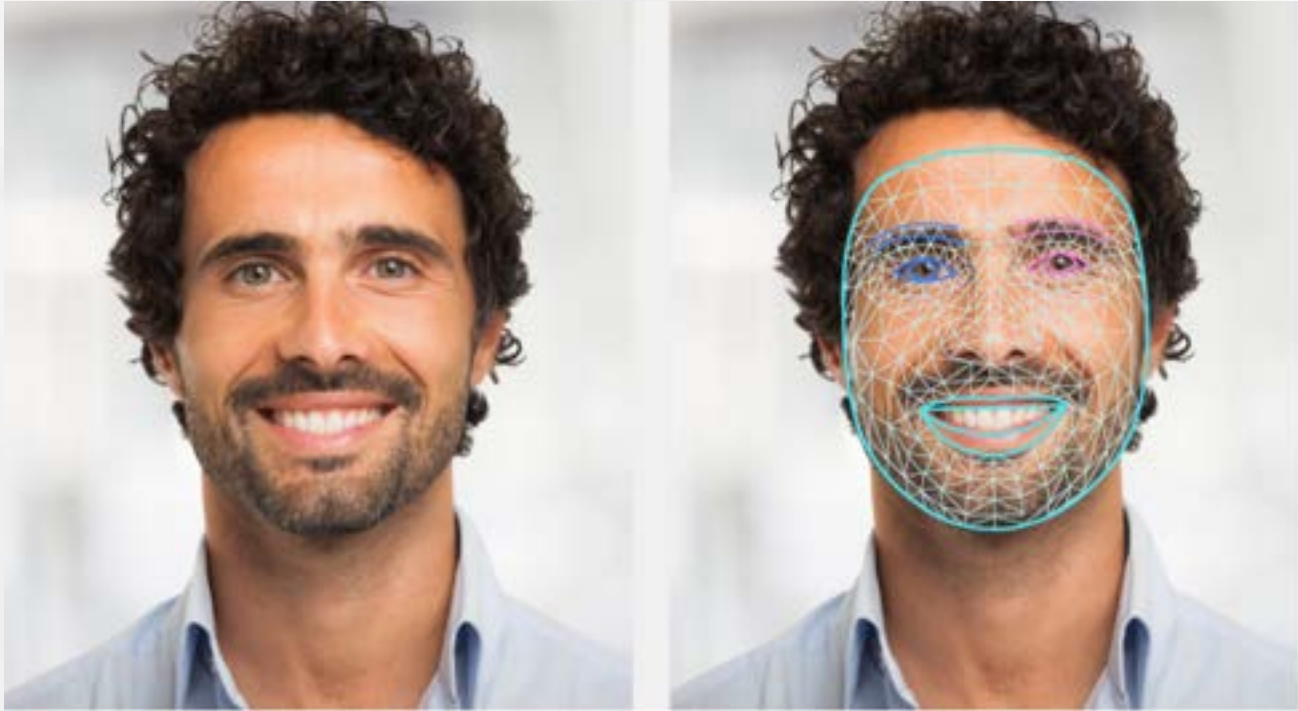
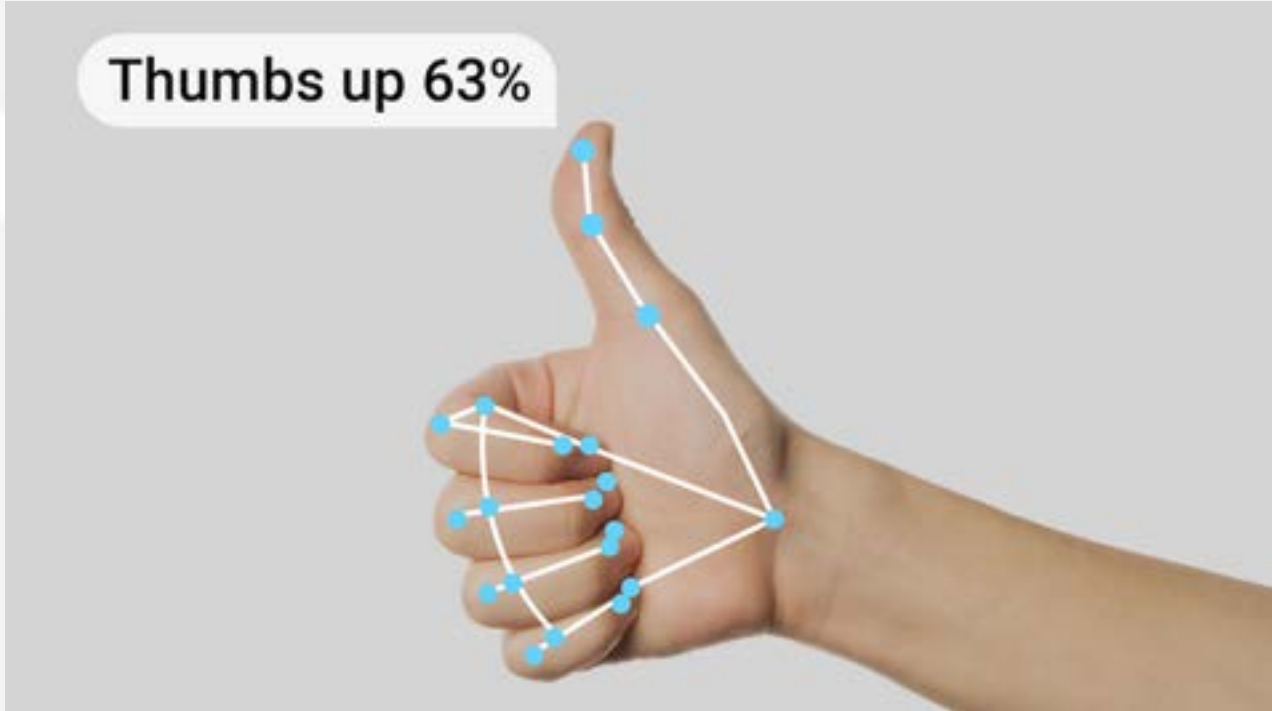
```
# Get coordinates
shoulder = [landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x,
            landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].y]
elbow = [landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x,
         landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].y]
wrist = [landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].x,
         landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].y]

# Calculate angle
angle = calculate_angle(shoulder, elbow, wrist)
```



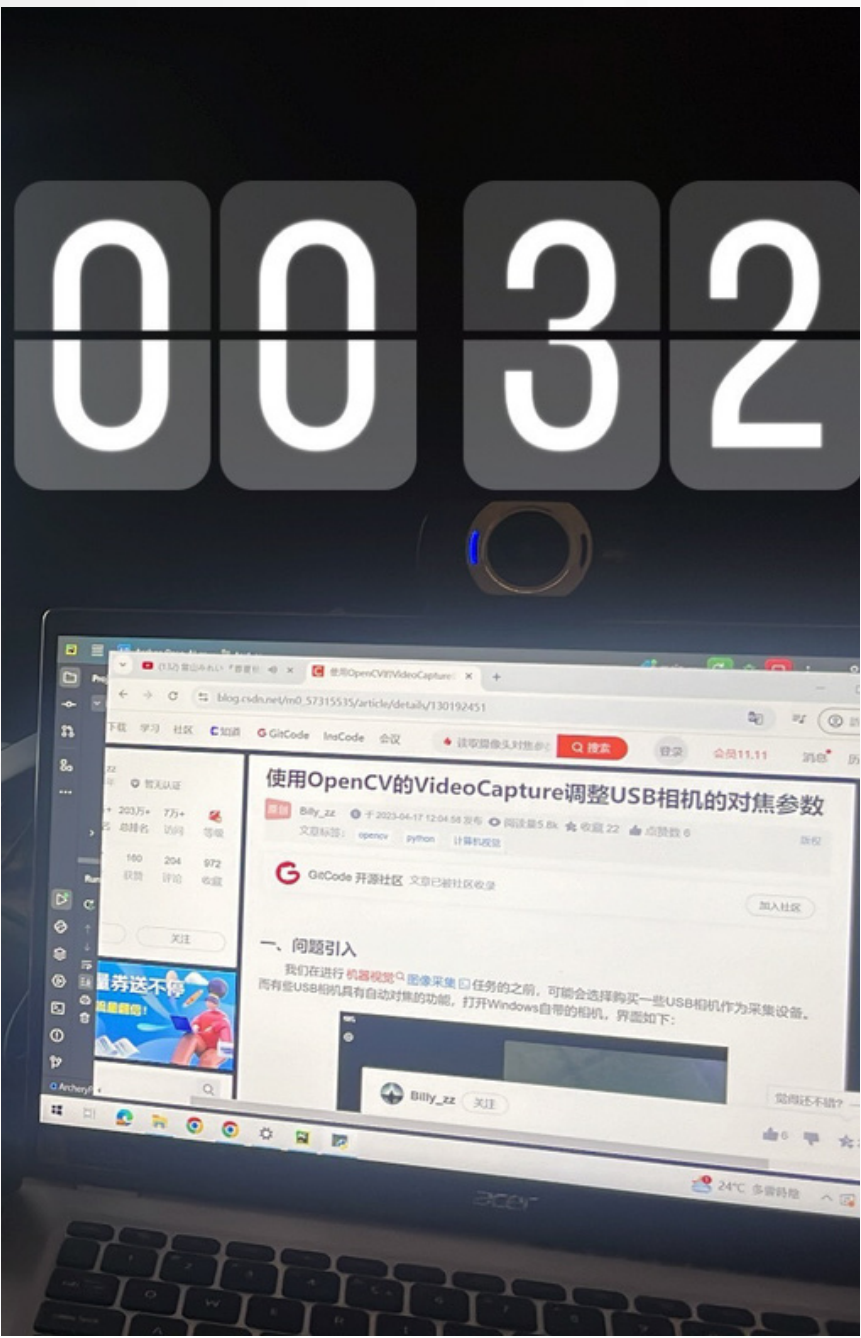
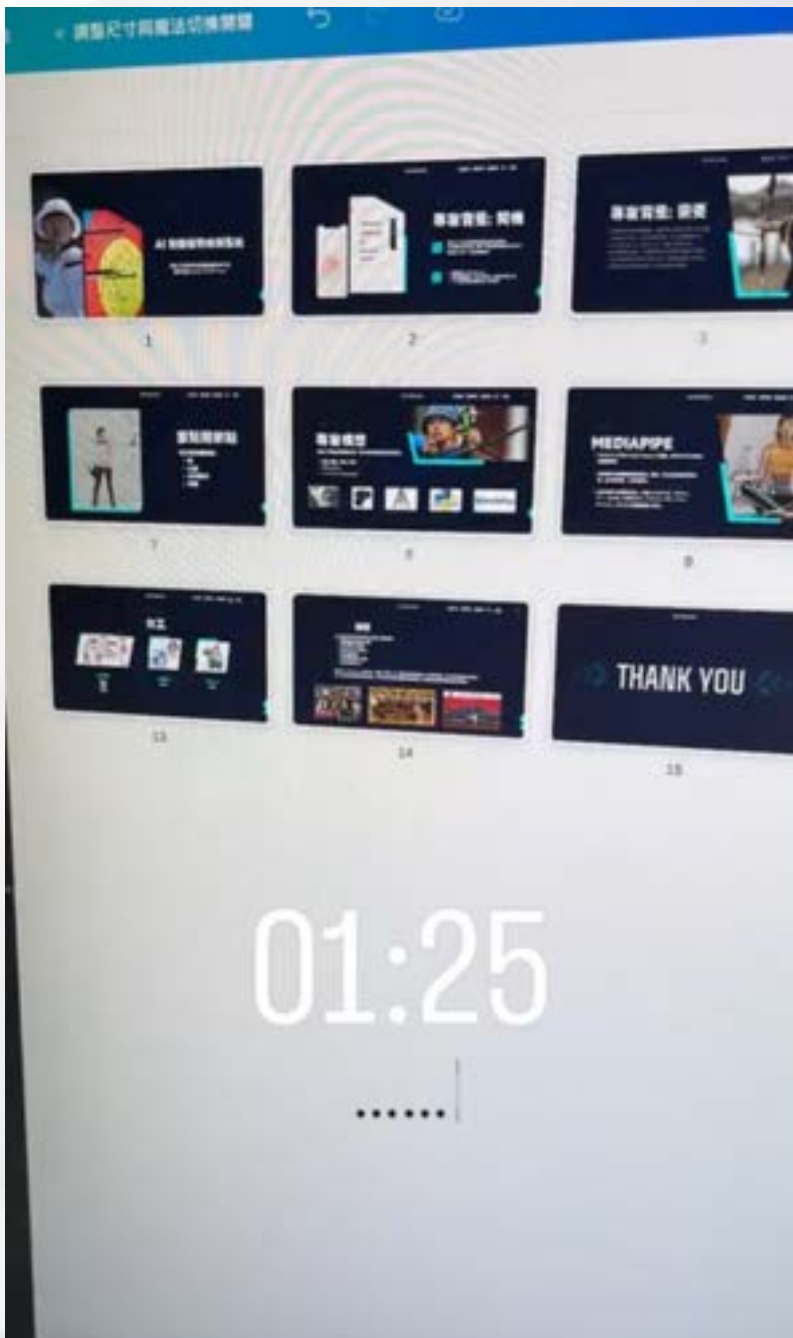
angle: absolute value of $\theta - \phi$

解決方案	Android	網頁版	Python	iOS	自訂模型
LLM Inference API	●	●		●	●
物件偵測	●	●	●	●	●
圖片分類	●	●	●	●	●
圖片區隔	●	●	●		
互動式區隔	●	●	●		
手部地標偵測	●	●	●	●	
手勢辨識	●	●	●	●	●
圖像嵌入	●	●	●		
臉部偵測	●	●	●	●	
臉部特徵偵測	●	●	●		
臉部造型	●	●	●		●
Pose 地標偵測	●	●	●		
圖像生成	●				●
文字分類	●	●	●	●	●
文字嵌入	●	●	●		
語言偵測器	●	●	●		
Audio classification	●	●	●		









總結

在中學籌劃及推動 STEAM 教育的策略：

- 結合自身的興趣及強項去啟發學生
- 為團隊製作詳盡的時間表及分工表
- 專案盡量以校內能實際應用來擬定框架
- ***善用網上及開源資源***
- 給學生篩選自學內容及資源
- 在過程中增進自己原本的技能及見識
- 和學生一起成長
- 成為學生團隊中的先鋒

