嵌入式系統總整與實作

曾煜棋、吳昆儒

National Yang Ming Chiao Tung University

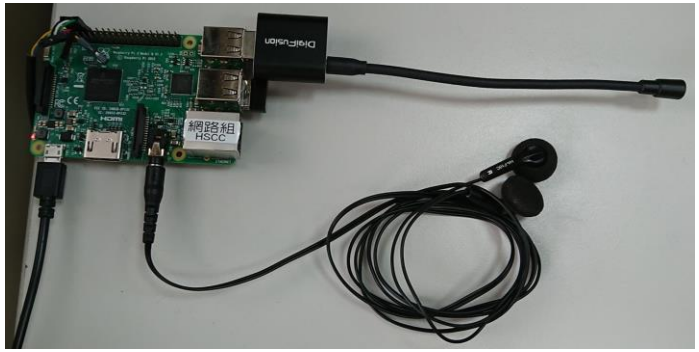
日期	主題
2/17	0. 課程介紹
2/24	梅竹賽!!
3/3	1. 嵌入式開發板 - 樹莓派介紹與設定 (headless)
3/10	2. 連接感測器 (GPIO, I2C) + 2 topic sharing
3/17	3. 處理感測資訊 (valuable data) + 2 topic sharing
3/24	4. 網路攝影機 IP cam+ 2 topic sharing
3/31	5. 語音互動
4/7	6.嵌入式 + AI模型: 邊緣裝置影像辨識
4/14	**期中考Midterm, Project分組**
4/21	專案檢索分享 (分組報告)
4/28	Final Project – Proposal (分組報告)
5/5	7. 嵌入式 + AI模型: 語音模型 (台灣樹莓派)
5/12	8. 網路應用: 推播廣告
5/19	9. 樹莓派核心編譯 (Cross compile, Kernel)
5/26	Final Project checkpoint, Q&A, 補demo
6/2	Final Project demonstration (學期考試周)
6/9, 16	彈性補充周

期中考周
(4/7-4/13)

期末考周
(5/27-6/2)

Last week

- 嵌入式應用: 語音識別 (語音助理)
  1. Mel-Frequency Cepstral Coefficients
  2. Speech to text (STT)
  3. Text to speech (TTS)

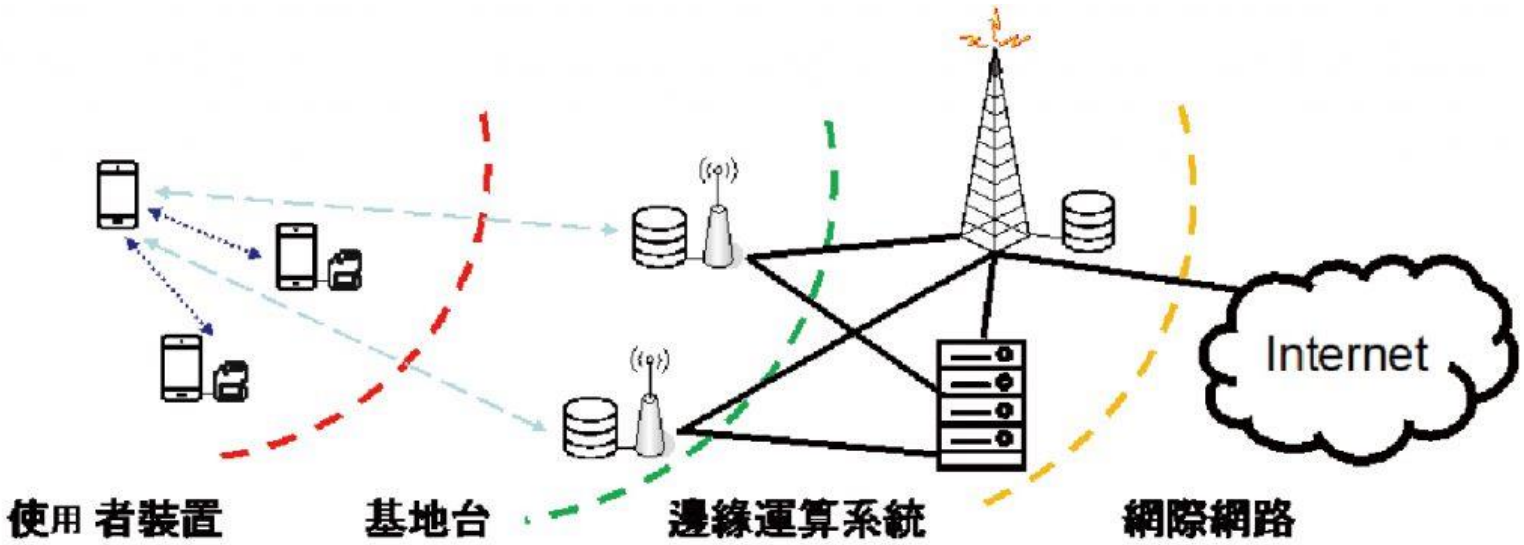


- 語音識別 (Speech recognition)
  - 自動語音辨識 (Automatic Speech Recognition, ASR)
  - 電腦語音識別 (Computer Speech Recognition)
  - 語音轉文字識別 (Speech To Text, STT)
  - 自然語言處理 (Natural Language Processing, NLP)
  - 讓電腦擁有理解人類語言的能力

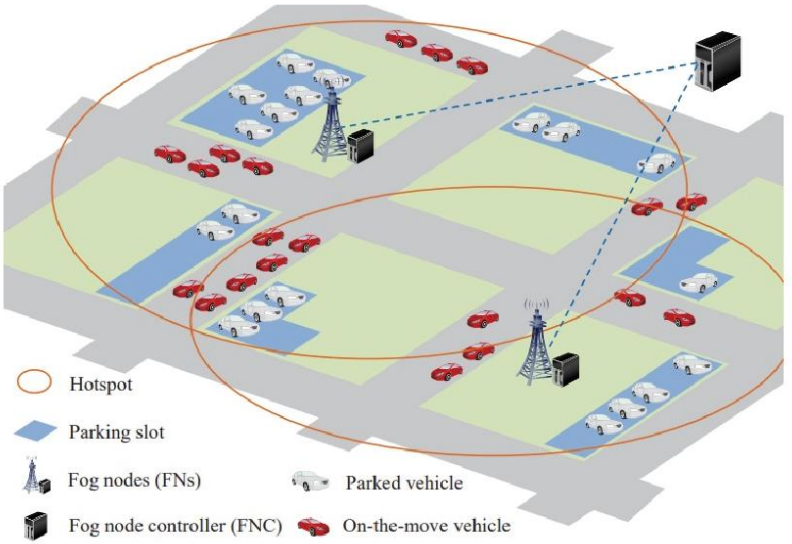
This week

- 嵌入式應用:嵌入式 + AI模型: 邊緣裝置影像辨識
  - 邊緣運算
  - 物件辨識 (object detection)
    - Tensorflow lite
  - 姿態識別 (pose estimation)
    - MediaPipe

邊緣運算



邊緣運算的基本精神為將有低延遲需求的計算服務從雲端網路轉移到網路邊緣，藉此降低實際傳輸距離、消除網路中不必要的資料傳輸及避免無法預測之網路壅塞，以滿足服務的低延遲需求。



https://newsletter.sinica.edu.tw/26211/

邊緣運算相關應用

- AWS邊緣運算服務
  - https://aws.amazon.com/tw/what-is/edge-computing/
- 晶睿推出邊緣運算人臉辨識攝影機
  - https://money.udn.com/money/story/5612/7061412
- ASUS將攝影機 AI 引進邊緣裝置
  - https://iot.asus.com/tw/news/Kdi3YqnS/

Object detection

- Google provides a sample quantized **SSDLite-MobileNet-v2** object detection model which is trained off the **MSCOCO dataset** and converted to run on **TensorFlow Lite**. It can detect and identify 80 different common objects, such as people, cars, cups, etc.

https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/blob/master/deploy_guides/Raspberry_Pi_Guide.md

https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html
https://arxiv.org/abs/1801.04381v4

SSDLite-MobileNet-v2

- MobileNetV2 is a very effective feature extractor for object detection and segmentation. For example, for detection when paired with the newly introduced SSDLite [2] the new model is about 35% faster with the same accuracy than MobileNetV1.

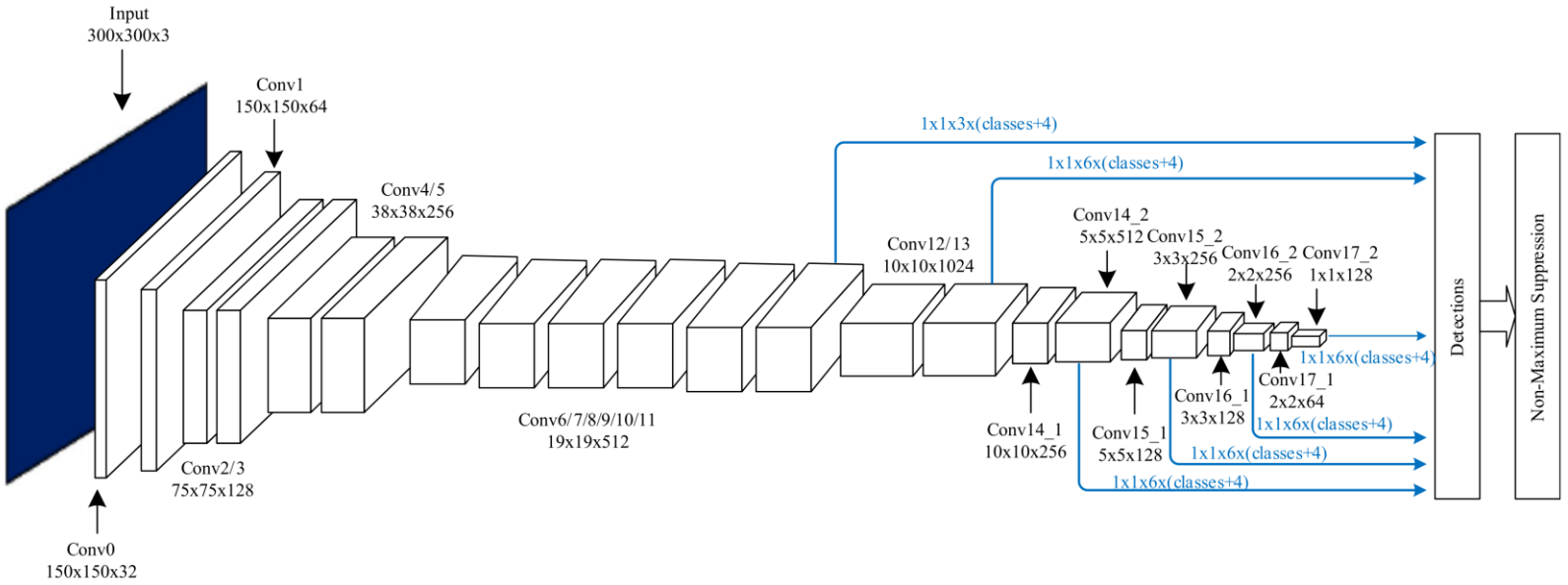
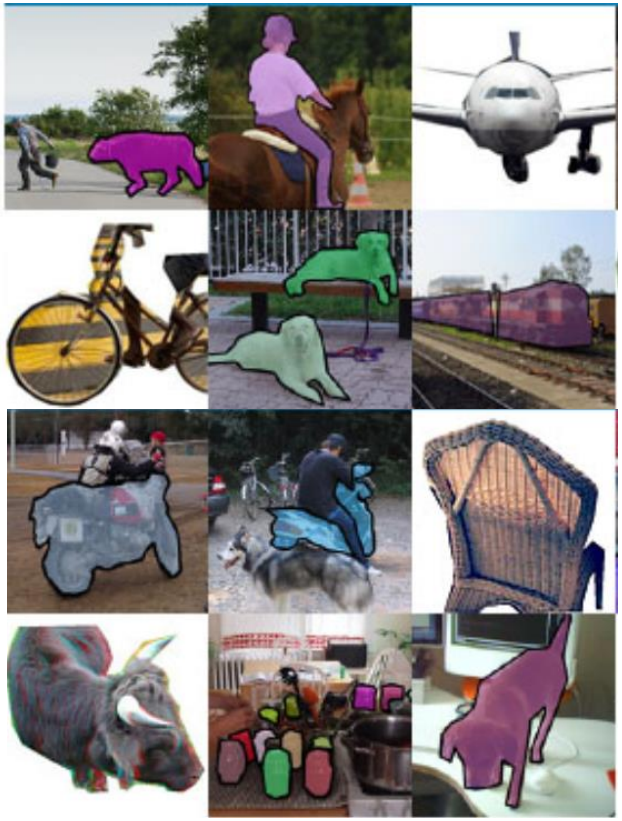


Fig source: https://link.springer.com/article/10.1007/s11042-022-13058-w

https://cocodataset.org/#home

MSCOCO dataset

- COCO is a large-scale object detection, segmentation, and captioning dataset.
- COCO has several features:
  - Object segmentation
  - Recognition in context
  - Superpixel stuff segmentation
  - 330K images (>200K labeled)
  - 1.5 million object instances
  - 80 object categories
  - 91 stuff categories
  - 5 captions per image
  - 250,000 people with keypoints

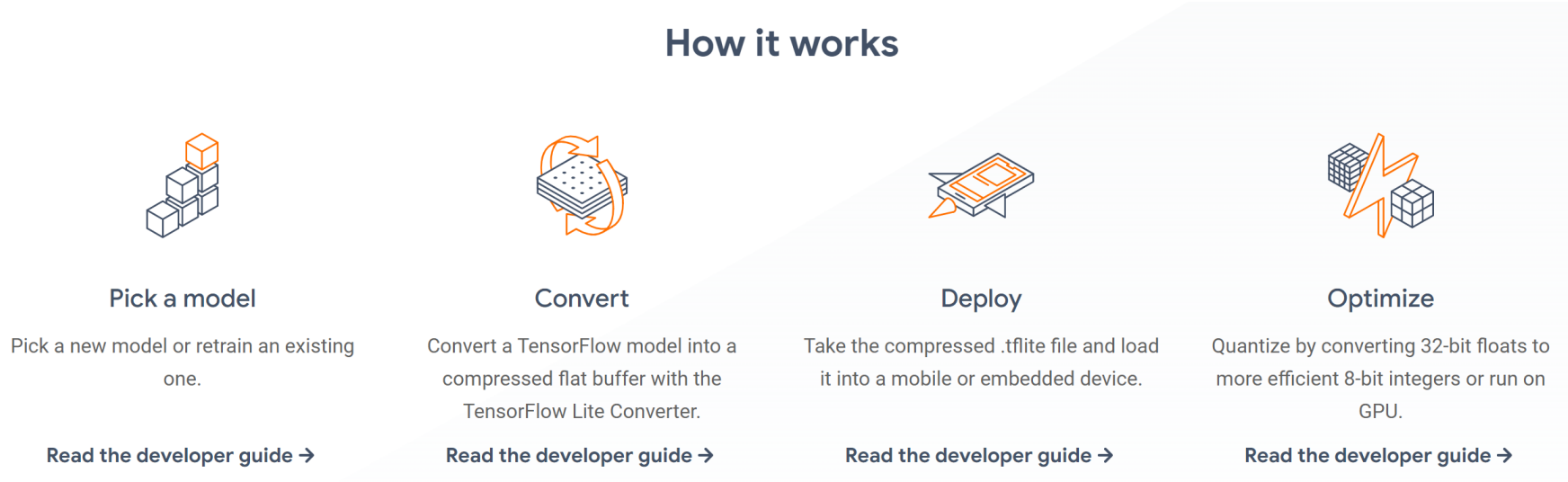


Dataset example

https://www.tensorflow.org/lite

TensorFlow Lite

- TensorFlow Lite is a mobile library for deploying models on mobile, microcontrollers and other edge devices.



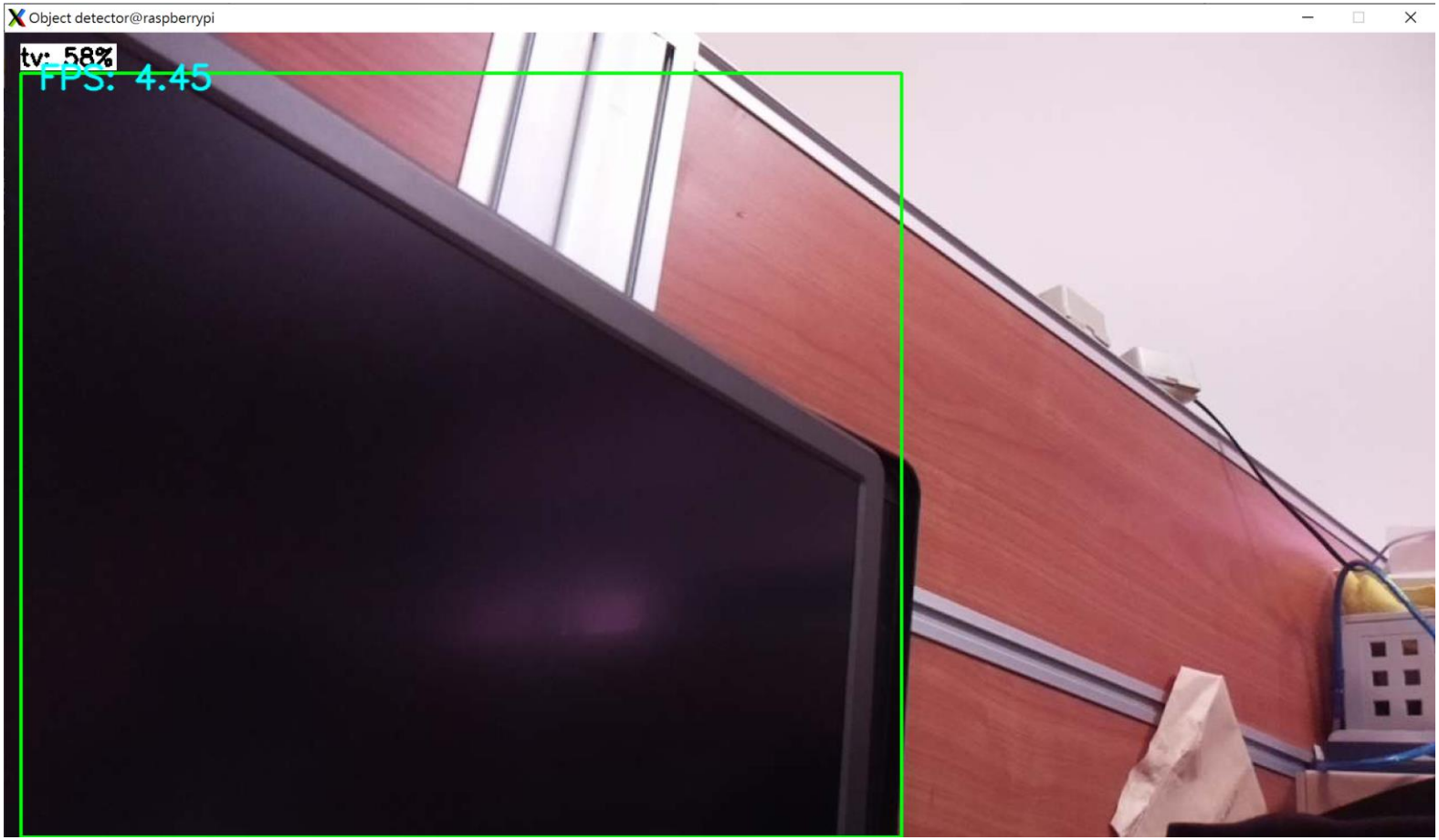
Build TensorFlow Lite Object Detection

- git clone https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi.git
- mv TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi tflite
- cd tflite/
- bash get_pi_requirements.sh
- wget https://storage.googleapis.com/download.tensorflow.org/models/tflite/coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip
- unzip coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip -d Sample_TFLite_model
- python3 TFLite_detection_webcam.py --modeldir=Sample_TFLite_model

TFLite_detection_webcam.py

- Initialize video stream & frame rate calculation
- for frame1 in camera.capture_continuous()
  - Start timer (for calculating frame rate
  - Grab frame from video stream
  - Acquire frame and resize
  - Normalize pixel values
  - Perform the actual detection by running the model
  - Retrieve detection results
  - Loop over all detections and draw detection box
    - draw bounding box, label and scores

Result



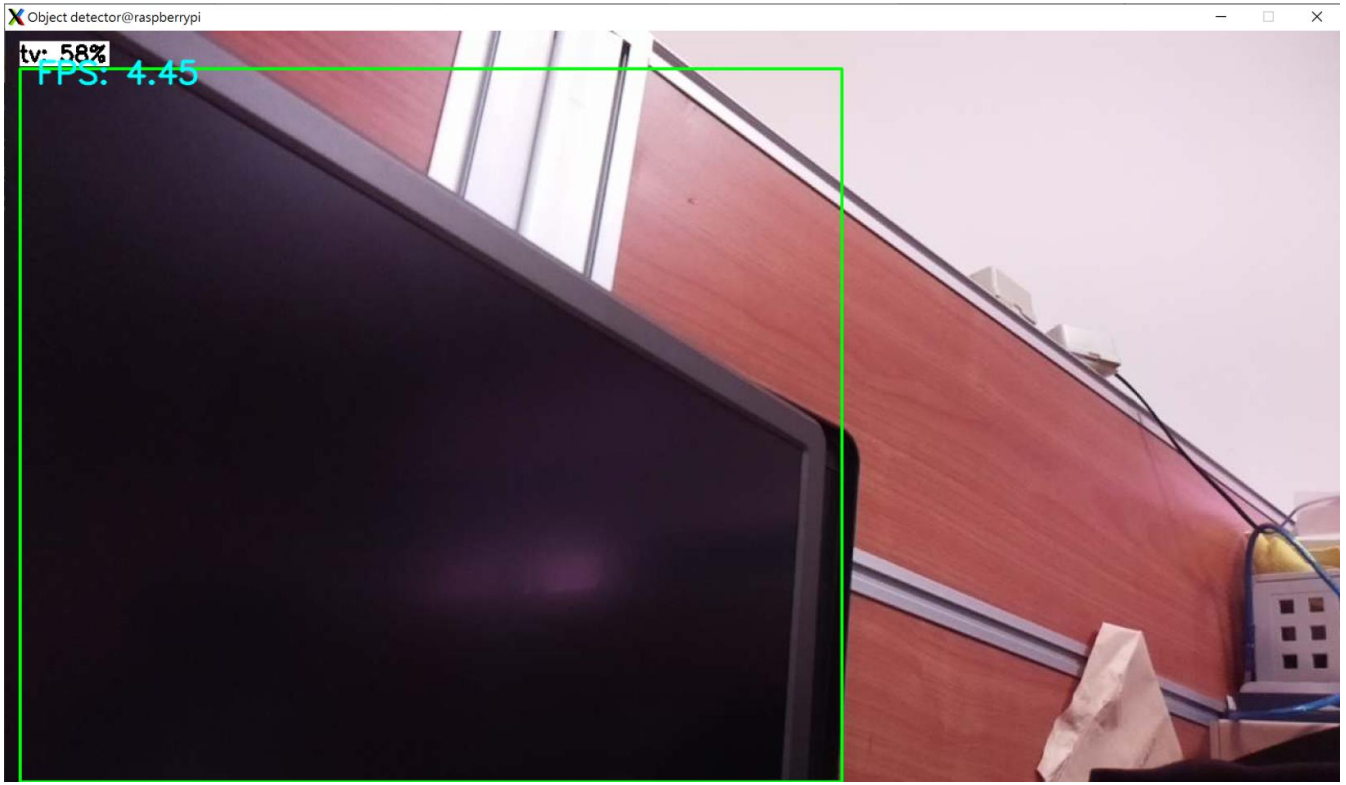
Lablemap info.

1	???	sheep	baseball glove	donut	toaster
2	person	cow	skateboard	cake	sink
3	bicycle	elephant	surfboard	chair	refrigerator
4	car	bear	tennis racket	couch	???
5	motorcycle	zebra	bottle	potted plant	book
6	airplane	giraffe	???	bed	clock
7	bus	???	wine glass	???	vase
8	train	backpack	cup	dining table	scissors
9	truck	umbrella	fork	???	teddy bear
10	boat	???	knife	???	hair drier
11	traffic light	???	spoon	toilet	toothbrush
12	fire hydrant	handbag	bowl	???	
13	???	tie	banana	tv	
14	stop sign	suitcase	apple	laptop	
15	parking meter	frisbee	sandwich	mouse	
16	bench	skis	orange	remote	
17	bird	snowboard	broccoli	keyboard	
18	cat	sports ball	carrot	cell phone	
19	dog	kite	hot dog	microwave	
20	horse	baseball bat	pizza	oven	

Under Sample_TFLite_model, you can find labelmap.txt

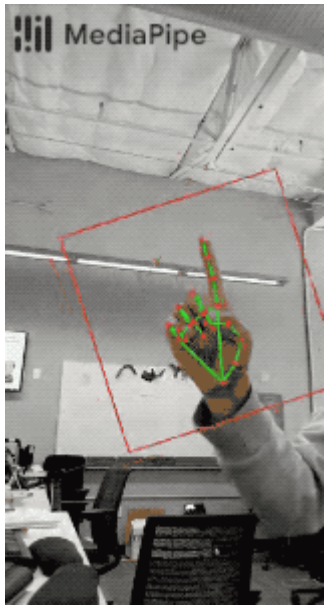
Discussion 1

- The object detection is very slow on Raspberry PI.
- Do you have any idea to improve the speed?

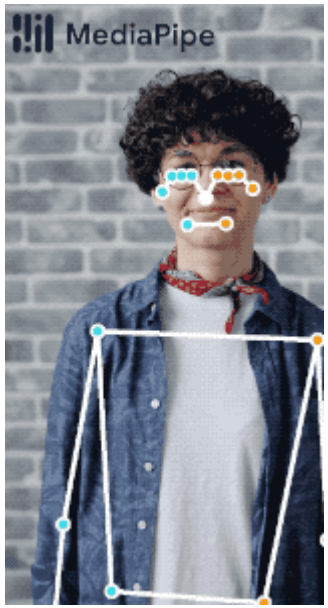


MediaPipe

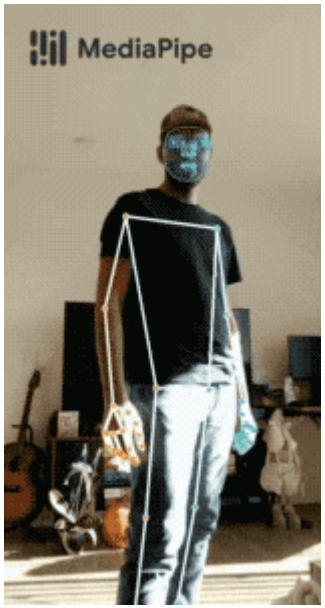
- MediaPipe offers cross-platform, customizable ML solutions for live and streaming media.



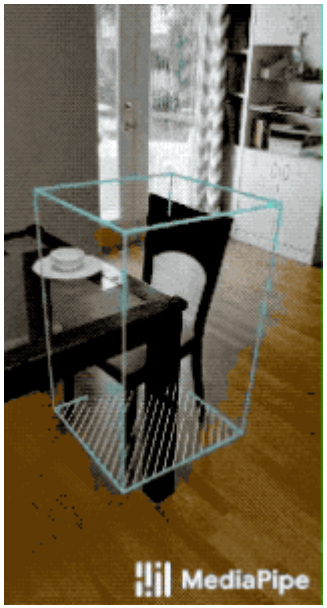
Hands



Pose



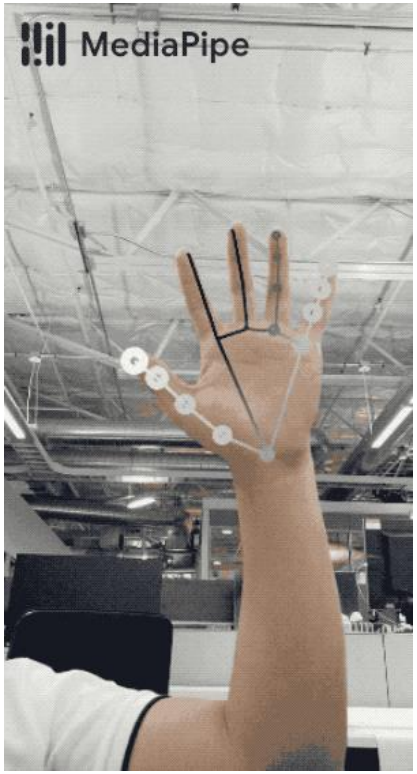
Holistic



Objectron

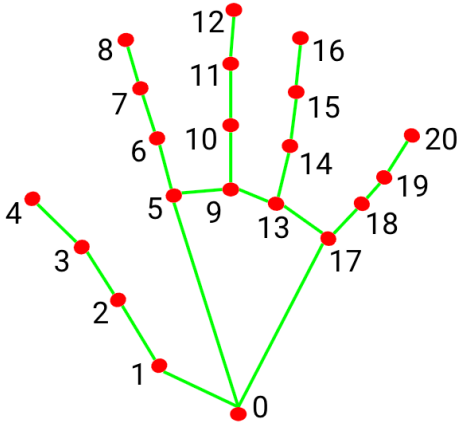
Build mediapipe

- pip3 install mediapipe-rpi4
- # two sample code: (1) Hand and (2) pose

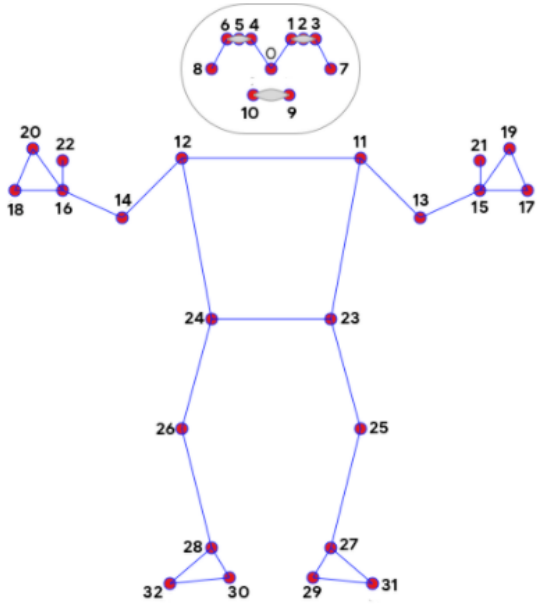


https://github.com/google/mediapipe/blob/master/docs/solutions/

Landmark Model



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Mediapipe model

- **Hands**
  - Palm detection model: TFLite model (lite), TFLite model (full)
  - Hand landmark model: TFLite model (lite), TFLite model (full)
  - Model card

- **Pose**
  - Pose detection model: TFLite model
  - Pose landmark model: TFLite model (lite), TFLite model (full), TFLite model (heavy)
  - Model card

Configuration Options

https://github.com/google/mediapipe/blob/master/docs/solutions/pose.md

- For pose detection:
  - model_complexity
    - Complexity of the pose landmark model: 0, 1 or 2. Landmark accuracy as well as inference latency generally go up with the model complexity. Default to 1.

- For hand detection:
  - model_complexity
    - Complexity of the hand landmark model: 0 or 1. Landmark accuracy as well as inference latency generally go up with the model complexity. Default to 1.

Discussion 2

- Try to use different model (lite, full and heavy) to run pose detection
  - mp.solutions.pose: model_complexity=??
- Capture the results for these three methods

- You can capture video or read a image
  - To read image, you might need: cv2.imread('xxx.jpg')

Error message?

```
  File "mp_pose.py", line 12, in <module>
    pose=mp_pose.Pose(model_complexity=0)
  File "/usr/local/lib/python3.7/dist-packages/mediapipe/python/solutions/pose.p
y", line 149, in __init__
    _download_oss_pose_landmark_model(model_complexity)
  File "/usr/local/lib/python3.7/dist-packages/mediapipe/python/solutions/pose.p
y", line 98, in _download_oss_pose_landmark_model
    'mediapipe/modules/pose_landmark/pose_landmark_lite.tflite')
  File "/usr/local/lib/python3.7/dist-packages/mediapipe/python/solutions/downlo
ad_utils.py", line 33, in download_oss_model
    with urllib.request.urlopen(model_url) as response, open(model_abspath,
  File "/usr/lib/python3.7/urllib/request.py", line 222, in urlopen
    return opener.open(url, data, timeout)
  File "/usr/lib/python3.7/urllib/request.py", line 531, in open
    response = meth(req, response)
  File "/usr/lib/python3.7/urllib/request.py", line 641, in http_response
    'http', request, response, code, msg, hdrs)
  File "/usr/lib/python3.7/urllib/request.py", line 569, in error
    return self._call_chain(*args)
  File "/usr/lib/python3.7/urllib/request.py", line 503, in _call_chain
    result = func(*args)
  File "/usr/lib/python3.7/urllib/request.py", line 649, in http_error_default
    raise HTTPError(req.full_url, code, msg, hdrs, fp)
urllib.error.HTTPError: HTTP Error 404: Not Found
```

mediapipe-rpi4 is not official python package.

The official mediapipe has updated the model URLs, but the package use old URLs.

Manual Download model

- pose=mp_pose.Pose(model_complexity=2):
  - wget https://storage.googleapis.com/mediapipe-assets/pose_landmark_heavy.tflite
  - sudo mv pose_landmark_heavy.tflite /usr/local/lib/python3.7/dist-packages/mediapipe/modules/pose_landmark/

- pose=mp_pose.Pose(model_complexity=0):
  - wget https://storage.googleapis.com/mediapipe-assets/pose_landmark_lite.tflite
  - sudo mv pose_landmark_lite.tflite /usr/local/lib/python3.7/dist-packages/mediapipe/modules/pose_landmark/

Summary

- Labs: PI camera + AI model
- Write down the answer for discussion, upload to e-campus. Deadline (before next class): 13:10, 4/14(Fri.)
  - Discussion1: How to improve the speed of object detection?
  - Discussion2: Try to use different model (lite, full and heavy) to run pose detection
  - 書面問答, 請上傳到e3
- Next week: Midterm

0414 - Midterm

- 筆試, 不用上機現場寫code
  - 題目會包含debug code, 可自行選擇帶零件過來
- Open book, open Internet
  - 可上網查詢資料, 也可看課程的相關講義檔案
  - 可以使用桌機, 筆電, 平板
  - 請勿使用手機, 禁止通訊軟體, 協作文件的comment...等
  - 所有可以交談的方式皆為禁止項目
    - 也不能問ChatGPT
- 考試時間13:20 – 16:10, 地點EC221, EC222
  - 座位之間須留空位
- 考試範圍: 上課提過的東西

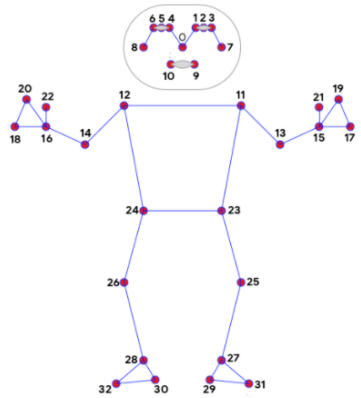
補充資料

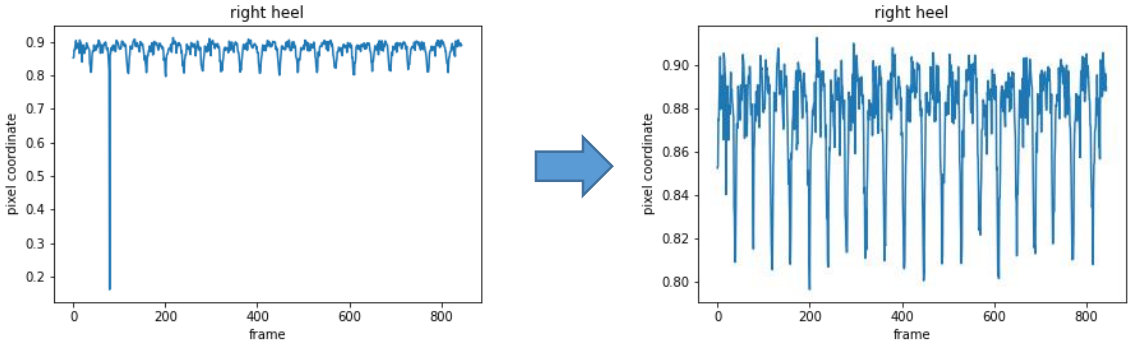
- A. 姿態識別結果的分析
- B. 姿態相似度分析
- C. Deepspeech 語音模型
- D. Virtualenv 虛擬環境

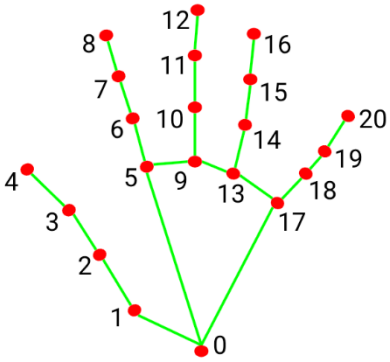
A. 計算指標

- 各項動作的共同指標
  - AI模型分析: 取得人體的關節點資訊
  - 擷取特定關節點的活動資訊 (如: 食指、大拇指、腳尖、肩膀...等)
  - 整理關節點的活動軌跡, 套用濾波器、計算波峰波谷位置, 得到相關的動作指標

  - 次數: 計算波形圖中的波峰出現次數
  - 動作時間: 計算相鄰波峰間的時間差
  - 完成時間: 第一個 – 最後一個動作的時間間隔
  - 頻率: 動作次數 / 整體動作時間
  - 震幅: 計算關節點的像素位移量
  - 規律性：完成每次動作的時間標準差



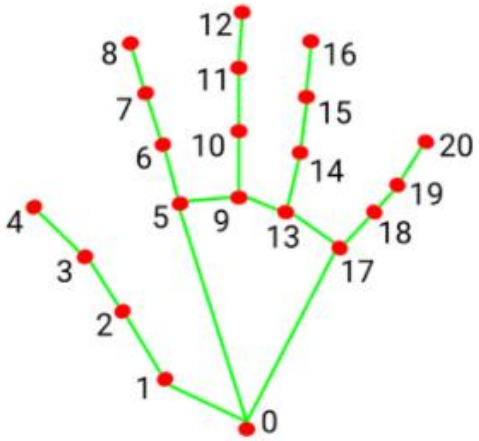




A. Use case: UPDRS

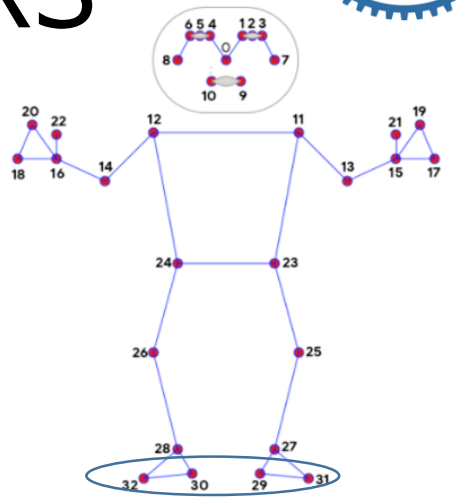
- 21. 靜止型顫抖
- 影像需求: 正面影像(上半身或下半身皆可，必須包含手部影像)
- 分析方式:
  - 1. 抓出手部骨架點，分別計算各關節點在X、Y方向上的移動軌跡
  - 2. 將移動軌跡圖輸入至FFT來獲得相應的顫抖頻率
  - 3. 選取介於3~8 Hz的domain frequency來當作此關節點的顫抖頻率

```
 0. 4.6075, 4.6075
 1. 4.6075, 0.8191
 2. 0.1024, 4.6075
 3. 0.1024, 4.6075
 4. 0.1024, 4.6075
 5. 0.1024, 0.2048
 6. 0.1024, 0.2048
 7. 4.6075, 0.2048
 8. 4.6075, 4.6075
 9. 0.1024, 4.6075
10. 4.6075, 4.6075
11. 4.6075, 4.6075
12. 4.6075, 4.6075
13. 0.1024, 4.6075
14. 4.6075, 4.6075
15. 4.6075, 4.6075
16. 4.6075, 4.6075
17. 0.1024, 4.6075
18. 4.6075, 4.6075
```

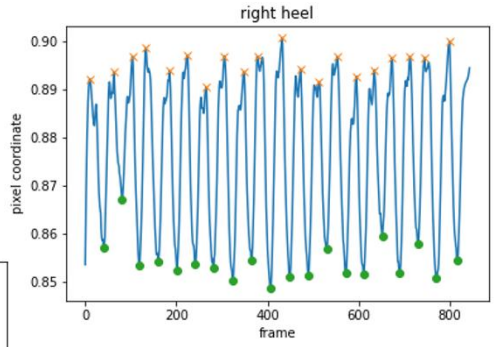


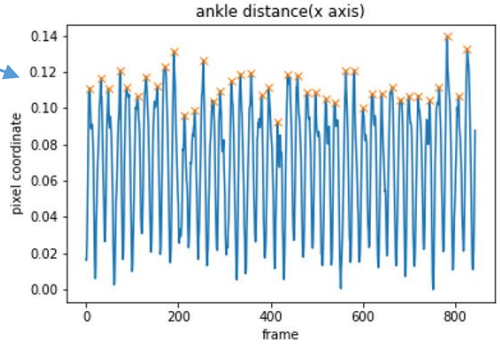
0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

A. Use case: UPDRS

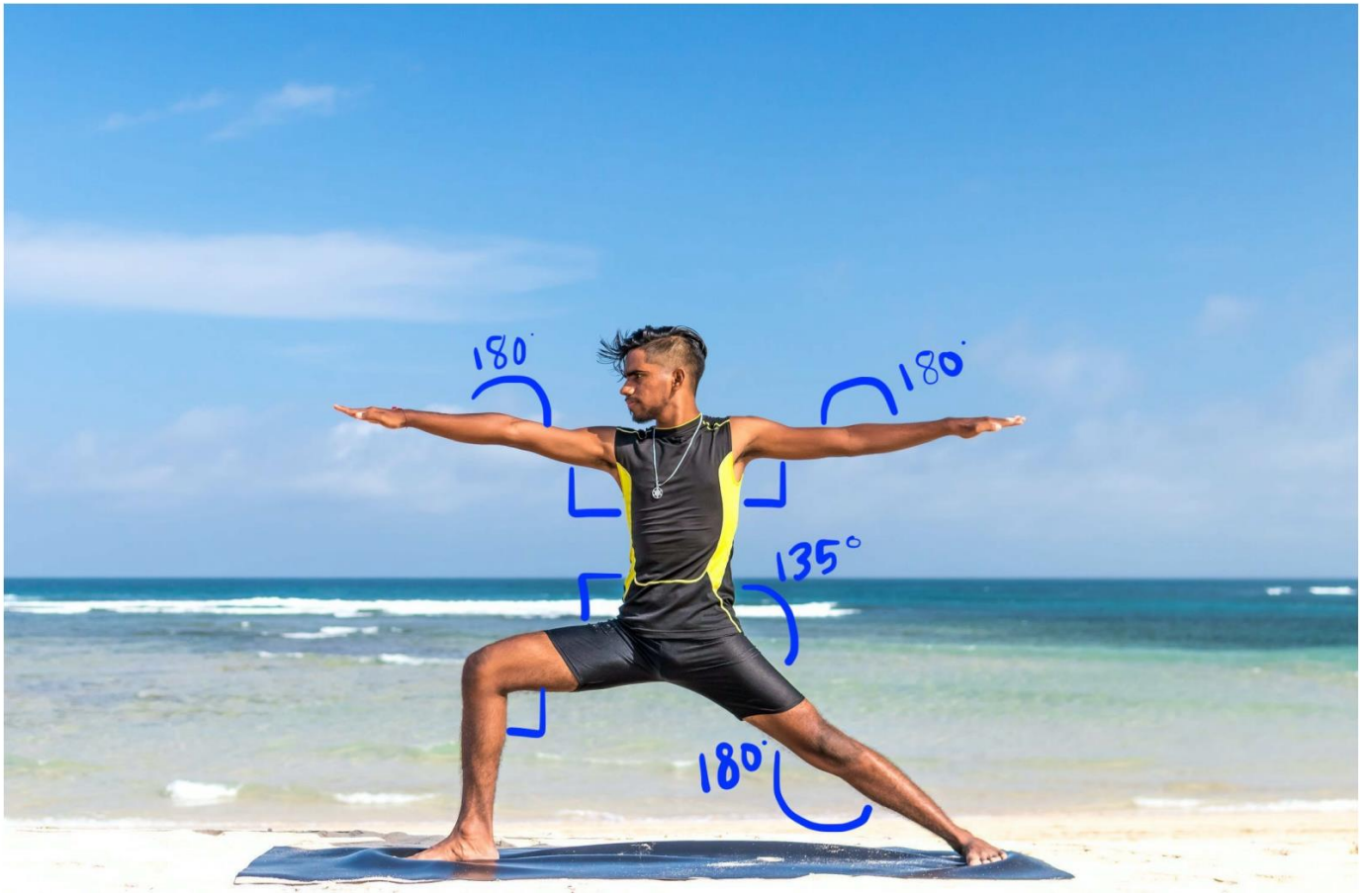


- 29. 步態
- 影像需求: 全身側面影像
- 分析方式:
  - 1. 獲得全身骨架點
  - 2. 擷取左腳根或右腳根在Y軸上的移動軌跡(計算步高)
  - 3. 擷取左腳踝和右腳踝在X軸方向上的間隔距離(計算步幅和步頻)
  - 3. 對上述波形做平滑化(Smoothing)
  - 4. 計算相應的參數
    - 步高: 計算移動軌跡波形圖的波谷震幅
    - 步幅: 波峰的震幅
    - 行走步數: 波峰的出現次數





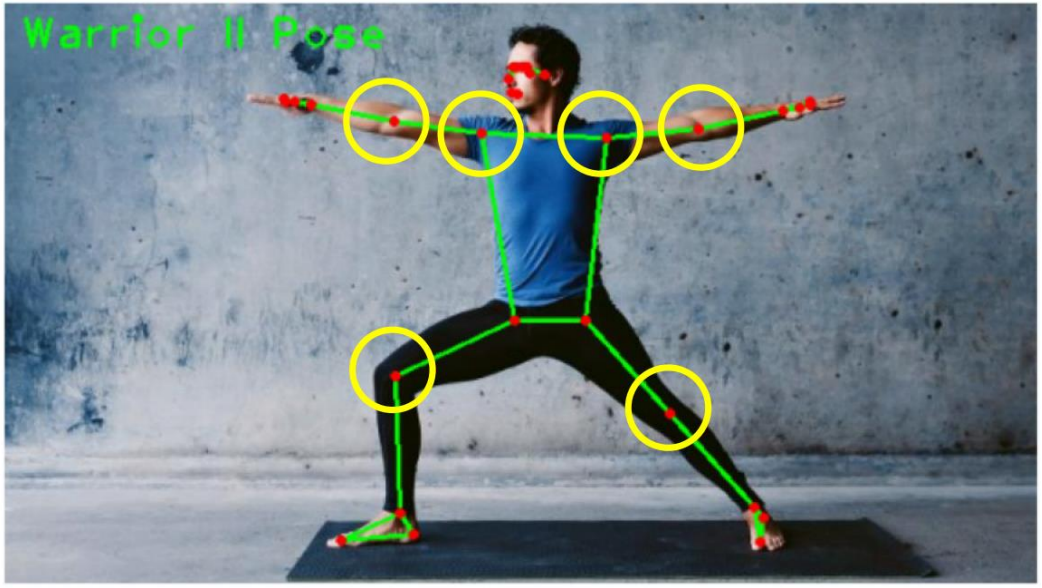
B. 姿態相似度分析



https://developers.google.com/ml-kit/vision/pose-detection/classifying-poses

B. 姿態相似度分析 (cont.)

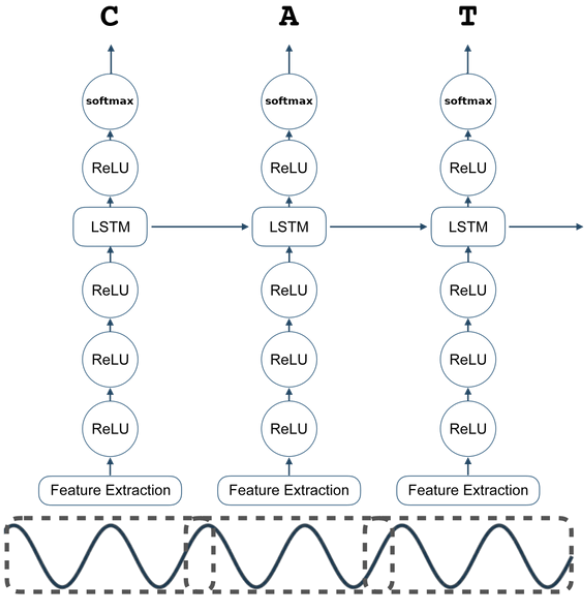
- 計算多個關鍵點間的角度 + 條件判斷
  - 195 > left_elbow_angle > 165,
  - 110 > left_shoulder_angle > 80,
  - 195 > left_knee_angle > 165,
  - 120 > right_knee_angle > 90



C. Deepspeech

- DeepSpeech is an open source embedded (offline, on-device) speech-to-text engine which can run in real time on devices ranging from a Raspberry Pi 4 to high power GPU servers.

On **Android** and **Raspberry Pi**, we only publish **TensorFlow Lite** enabled packages, and they are simply called deepspeech



The complete RNN model

D. Virtualenv

- virtualenv is a tool to create isolated Python environments

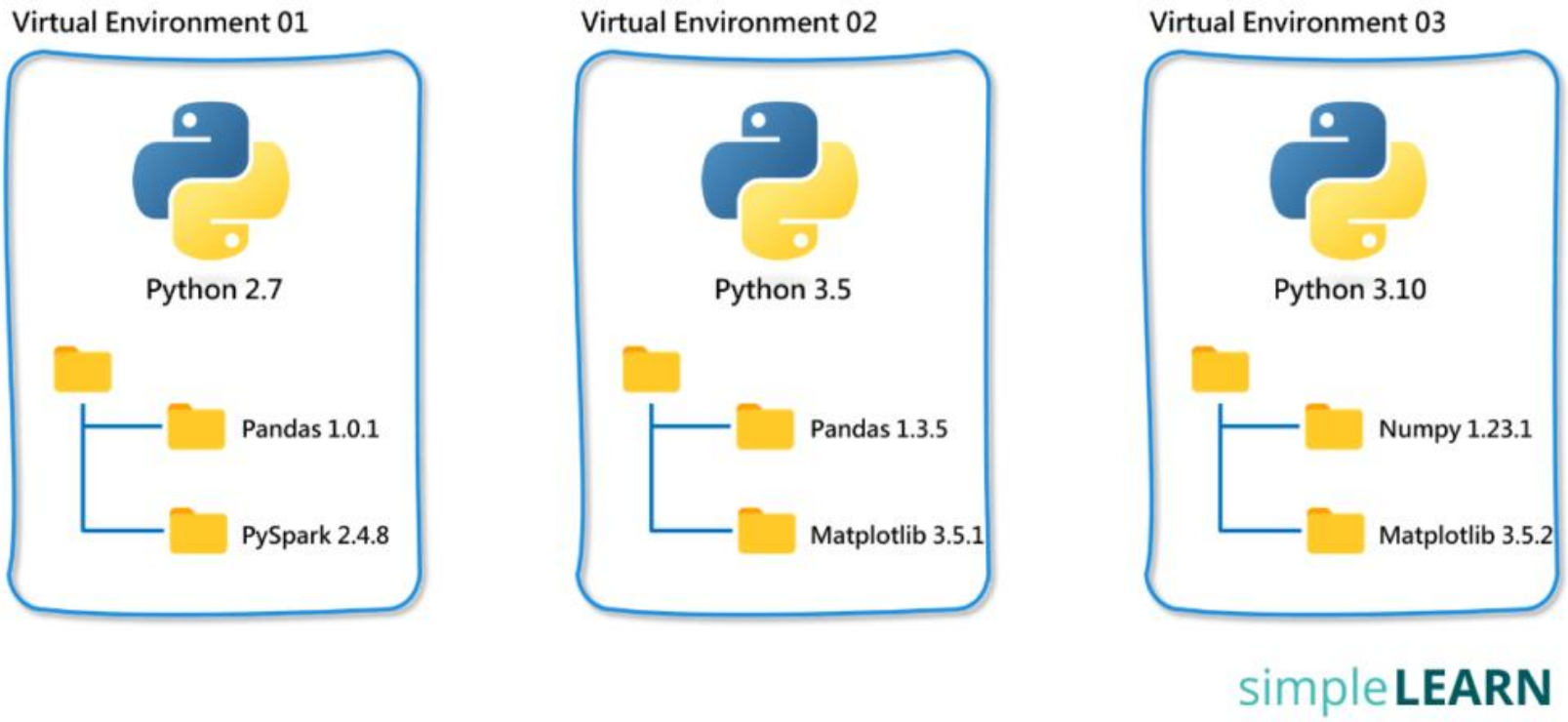


Fig source: https://simplelearn.tw/2022/08/08/ python-虛擬環境完整教學-1｜virtualenv-虛擬環境安裝及使用/

D. Command: pip list

- List installed packages, including editables.
- Packages are listed in a case-insensitive sorted order.

```
pi@raspberrypi:~ $ pip3 list
Package                 Version
----------------------  -----------
absl-py                 1.4.0
anyio                   3.6.2
argon2-cffi             21.3.0
argon2-cffi-bindings    21.2.0
…
…
wheel                   0.40.0
widgetsnbextension      4.0.7
wrapt                   1.10.11
zipp                    3.15.0
(# end)
```

D. Virtualenv

- List installed packages under a new virtualenv

```
pi@raspberrypi:~ $ source tflite-env/bin/activate
(tflite-env) pi@raspberrypi:~ $ pip3 list
Package            Version
-------------      -------
pip                18.1
pkg-resources      0.0.0
setuptools         40.8.0
```

- Leave command: deactivate

```
(tflite-env) pi@raspberrypi:~ $ deactivate
pi@raspberrypi:~ $
```