

— 瑜珈姿勢檢測

Yoga Pose Detection

Yoga Master

Instructor: 張志勇教授、蒯思齊教授

Team Leader: 047 劉建宏

Team Members:

014 陳亭均

015 梁毓琳

025 毛彥文

051 于復申

046 邵文綺



— Agenda

- Objective & Motivation
- Why Yoga Master
- Technical & Structure
- Optimization Evaluation
- Conclusion
- Reference



Objective & Motivation 緣起

歷經了三年新冠疫情，由於公共場所(如健身房等)因隔離政策暫時關閉的緣故，瑜珈成為愈來愈多人調整身心靈的運動方式。

Yoga Master的創立目的便是希望能推廣瑜珈運動，

利用 “動作行為辨識” + “檢測人體關鍵點” 不需去瑜珈教室也能有效地調整自己的瑜珈姿勢。

Yoga Master會針對下列五種瑜珈姿勢，進行辨認並讓使用者知悉自己的瑜珈姿勢的正確率。



Yoga Pose

About Yoga

瑜珈姿勢說明

下犬式-瑜珈動作的基本起始動作之一，做的時候身體會呈現一個倒 V 字型，可以用非常和緩的方式來伸展背部和腿部的肌肉，同時也可以順便訓練腹部和手臂肌肉，並幫助肩頸的放鬆。

女神式-雙腳打開。身體下蹲，將膝蓋彎曲，雙手打開往上伸直。

平板式-從下犬式開始，身體往前移動。雙腳往後退一小步，保持核心用力，完成平板式。

樹式-山式，雙腳併攏站立，雙腳內側相互觸碰，身體重量均勻地放在雙腳全腳掌上。雙膝併攏，大腿內側收緊，臀部收緊。收腹，挺胸，整條脊柱向上伸展拉長，頭、頸端正。雙肩下沉，手臂向下伸展。保持身體在這種狀態上不動。

勇士二式-腳掌踩地，身體抬起，臀部向上提起，身體呈現倒三角形。左腳向前跨置於雙手之間，右腳尖向外旋轉90度踩穩。雙臂打開與肩膀呈水平，左手向前延伸，右手向後延伸。左腳成弓箭步站姿，右腳膝蓋伸直，上身保持挺直。



Downward dog

下犬式



Goddess

女神式



Plank

平板式



Tree

樹式



Warrior2

勇士二式

Why Yoga Master

相關技術

Yoga Master的優勢

現存的解決方案

- 捲積神經網路對瑜珈動作分類
- 瑜珈動作的關鍵點檢測

現有方案的缺點及瓶頸

- 尚未有結合瑜珈動作分類與關鍵點檢測的模型

本專題的優勢

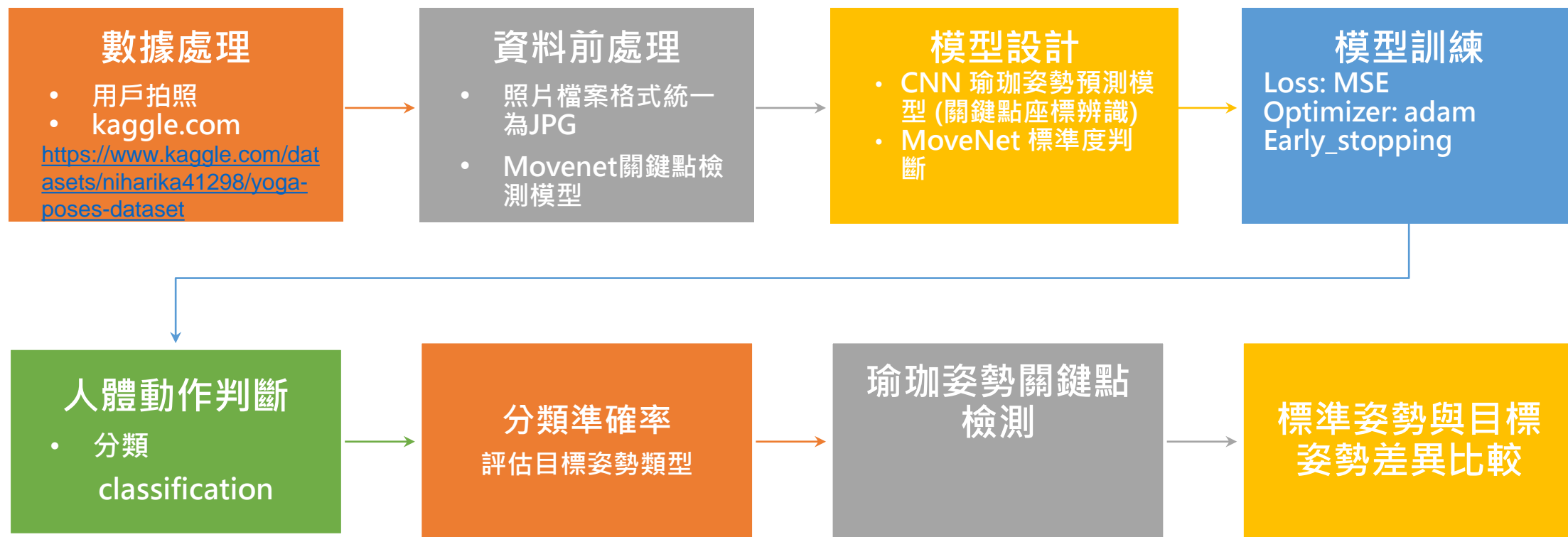
- 透過結合瑜珈動作分類&關鍵點檢測來對使用者的姿勢進行標準度判斷

Technical & Structure

系統架構及技術

yogamaster_overpower.ipynb

https://github.com/quilty1012/NYMCU/blob/main/yogamaster_overpower_r.ipynb



Technical & Structure

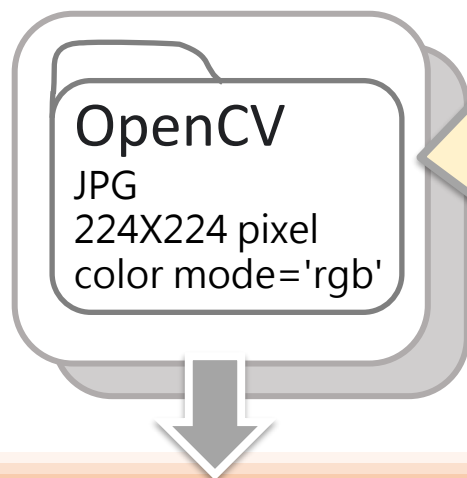
系統架構及技術

流程表

數據蒐集



數據期前處理1



統一照片檔案

```
def convert_to_jpg(input_folder, output_folder):  
    if not os.path.exists(output_folder): # 確保輸出資料夾存在  
        os.makedirs(output_folder)  
    file_list = os.listdir(input_folder) # 列出輸入資料夾中的所有檔案  
    for file_name in file_list:  
        name, ext = os.path.splitext(file_name) # 將檔案名稱和副檔名拆開  
        if ext.lower() in ['.png', '.jpeg', '.jpg']: # 確保是圖片檔案  
            img_path = os.path.join(input_folder, file_name) # 讀取圖片  
            img = cv2.imread(img_path)  
            output_path = os.path.join(output_folder, name + '.jpg') # 將圖片另存為JPG格式  
            cv2.imwrite(output_path, img)  
            print(f"Converted {file_name} to JPG.")  
input_folder = '../DATASET/TEST/downdog'  
output_folder = 'DATASET/TEST/downdog'  
convert_to_jpg(input_folder, output_folder) # 執行轉換
```


Technical & Structure

流程表

系統架構及技術

數據期前處理2

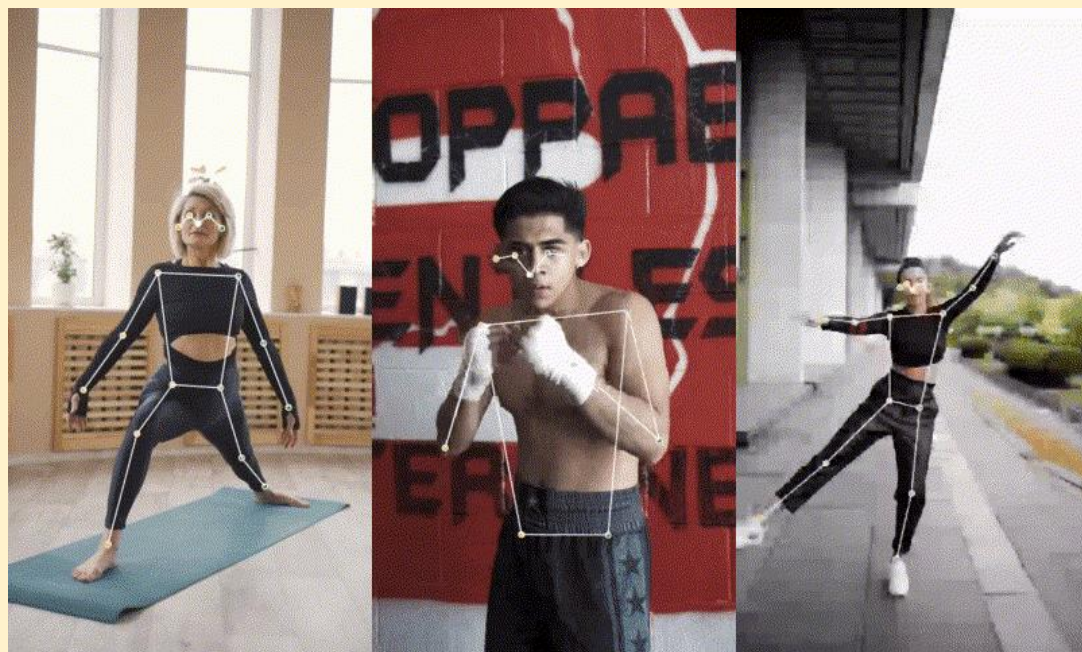
MoveNet

Ultra fast and accurate
pose detection model

模型設計

人體關鍵點檢測模型

Next-Generation Pose Detection with MoveNet and TensorFlow.js



<https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html>

Technical & Structure

系統架構及技術

數據期前處理2

MoveNet

Ultra fast and accurate
pose detection model

模型設計

身體的關鍵點 INDEX

```
KEYPOINT_DICT={  
    'nose' :0,  
    'left_eye' :1,  
    'right_eye' :2,  
    'left_ear' :3,  
    'right_ear' :4,  
    'left_shoulder' :5,  
    'right_shoulder' :6,  
    'left_elbow' :7,  
    'right_elbow' :8,  
    'left_wrist' :9,  
    'right_wrist':10,  
    'left_hip':11,  
    'right_hip':12,  
    'left_knee':13,  
    'right_knee':14,  
    'left_ankle':15,  
    'right_ankle':16 }
```



Technical & Structure

系統架構及技術

流程表

數據期前處理2

MoveNet

Ultra fast and accurate pose detection model

<https://www.tensorflow.org/hub/tutorials/movenet>

模型設計

MoveNet關鍵點檢測模型

```
sub_folders = ['downdog', 'goddess', 'plank', 'tree', 'warrior2'] # 子資料夾列表
movenet_keypoints = {} # 儲存關鍵點座標的字典
for sub_folder in sub_folders: # 遍歷子資料夾，處理圖片
    folder_path = os.path.join(train_dir, sub_folder)
    image_list = os.listdir(folder_path)
    for image_name in image_list:
        image_path = os.path.join(folder_path, image_name)
        image = load_img(image_path, target_size=(224, 224))
        input_image = tf.expand_dims(image, axis=0)
        input_image = tf.image.resize_with_pad(input_image, input_size, input_size)
        reshaped_data = movenet(input_image) # 使用MoveNet模型進行關鍵點檢測
        reshaped_data = reshaped_data.reshape(17, 3)
        keypoints=reshaped_data[:, :2]
        keypoints=np.around(keypoints,2)
        keypoints=keypoints.flatten()
        folder_name = os.path.basename(sub_folder)
        # 儲存關鍵點座標到字典中，子資料夾和檔案名稱作為鍵
        image_key = os.path.join(folder_name, image_name)
        movenet_keypoints[image_key] = keypoints
```

Technical & Structure

流程表

系統架構及技術

17個點，共34個座標

數據期前處理2

MoveNet

Ultra fast and accurate
pose detection model

<https://www.tensorflow.org/hub/tutorials/movenet>

模型設計

CNN

瑜珈姿勢預測模型
關鍵點座標辨識

MoveNet

標準度判斷

	nose_y	nose_x	left_eye_y	left_eye_x	right_eye_y	right_eye_x	left_ear_y	left_ear_x	right_ear_y	right_ear_x	...
downdog/00000005.jpg	0.66	0.26	0.67	0.24	0.67	0.24	0.63	0.21	0.63	0.21	...
downdog/00000003.jpg	0.75	0.54	0.76	0.55	0.76	0.55	0.73	0.61	0.73	0.61	...
downdog/00000000.jpg	0.73	0.52	0.74	0.51	0.74	0.51	0.71	0.45	0.71	0.45	...
downdog/00000006.jpg	0.76	0.39	0.76	0.38	0.76	0.38	0.70	0.36	0.70	0.35	...
downdog/00000002.jpg	0.72	0.59	0.73	0.61	0.73	0.62	0.68	0.66	0.68	0.66	...
...
warrior2/00000116.jpg	0.05	0.32	0.01	0.34	0.02	0.34	0.01	0.41	0.02	0.40	...
warrior2/00000117.jpg	0.17	0.39	0.15	0.42	0.15	0.41	0.17	0.47	0.17	0.46	...
warrior2/00000114.jpg	0.23	0.56	0.21	0.56	0.21	0.54	0.22	0.52	0.22	0.48	...
warrior2/00000108.jpg	0.05	0.32	0.01	0.34	0.02	0.34	0.01	0.41	0.02	0.40	...
warrior2/00000115.jpg	0.21	0.44	0.18	0.47	0.18	0.43	0.19	0.53	0.19	0.44	...
	right_hip_x	left_knee_y	left_knee_x	right_knee_y	right_knee_x	left_ankle_y	left_ankle_x	right_ankle_y	right_ankle_x	target	
downdog/00000005.jpg	0.41	0.57	0.51	0.57	0.50	0.78	0.58	0.77	0.57	0	
downdog/00000003.jpg	0.41	0.55	0.30	0.55	0.30	0.80	0.19	0.81	0.19	0	
downdog/00000000.jpg	0.66	0.57	0.78	0.58	0.78	0.86	0.88	0.84	0.86	0	
downdog/00000006.jpg	0.61	0.55	0.77	0.54	0.77	0.88	0.91	0.83	0.88	0	
downdog/00000002.jpg	0.39	0.56	0.23	0.56	0.22	0.82	0.10	0.83	0.09	0	
...	
warrior2/00000116.jpg	0.47	0.84	0.81	0.73	0.25	0.99	0.94	0.99	0.30	4	
warrior2/00000117.jpg	0.41	0.78	0.59	0.55	0.28	0.90	0.77	0.81	0.30	4	
warrior2/00000114.jpg	0.39	0.60	0.70	0.74	0.21	0.82	0.74	0.83	0.07	4	
warrior2/00000108.jpg	0.47	0.84	0.81	0.73	0.25	0.99	0.94	0.99	0.30	4	
warrior2/00000115.jpg	0.46	0.70	0.69	0.65	0.32	0.85	0.81	0.85	0.38	4	

Technical & Structure

系統架構及技術

流程表

模型設計

CNN
瑜珈姿勢預測模型
關鍵點座標辨識

MOVENET
標準度判斷

模型訓練

Loss: MSE

Optimizer:
adam

Early_stopping

模型設計

```
from keras.layers import Dense, Conv1D, Flatten, MaxPool1D
```

```
model = Sequential() #順序型模型類型，  
                    單一輸入，單一輸出
```

17個偵測點
=34個座標
(34個欄位)

```
model.add(Conv1D(16, 2, input_shape=(34,1), activation="relu"))
```

```
model.add(Conv1D(32, 2, activation = "relu"))
```

```
model.add(MaxPool1D())
```

```
model.add(Flatten())
```

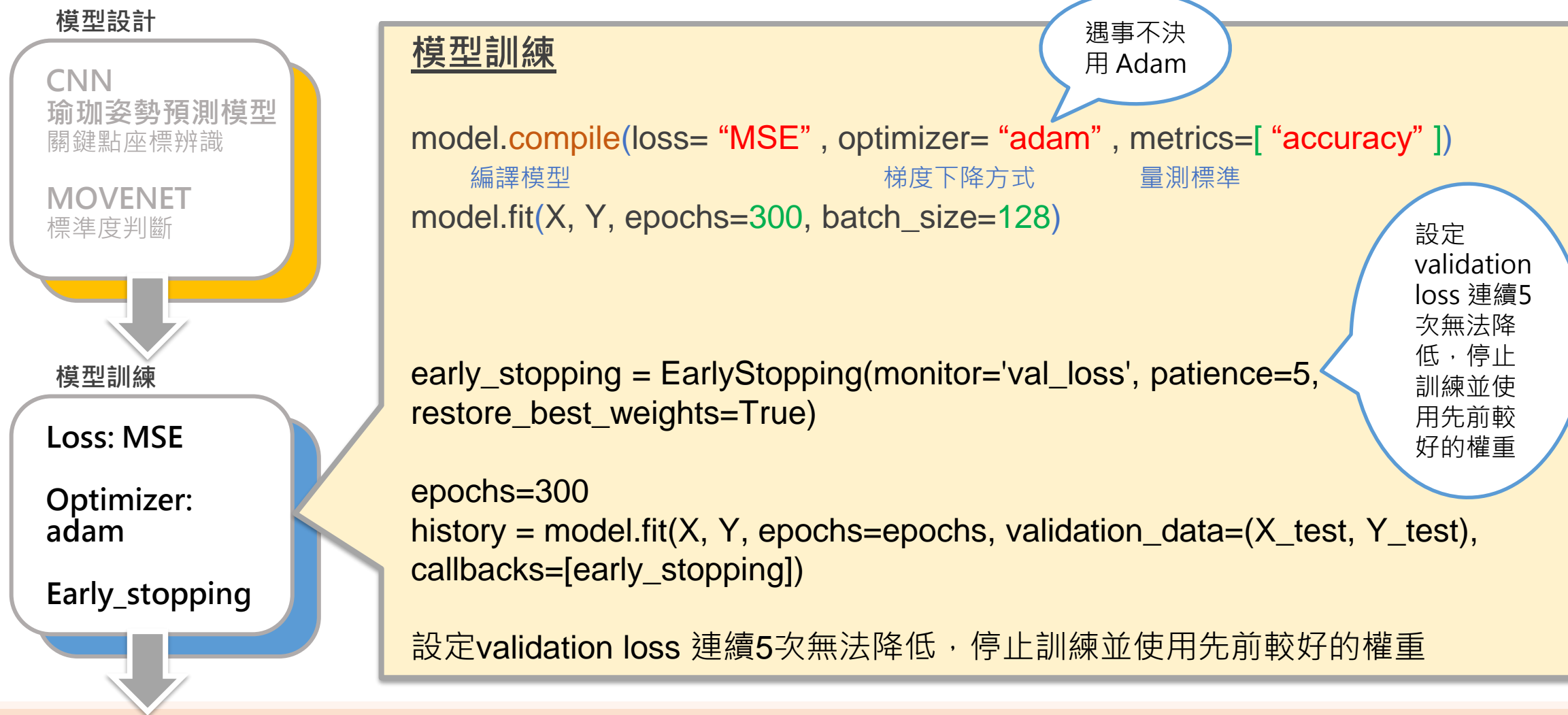
```
model.add(Dense(5, activation = "softmax" ))
```

```
model.summary()
```

Technical & Structure

流程表

系統架構及技術



Technical & Structure

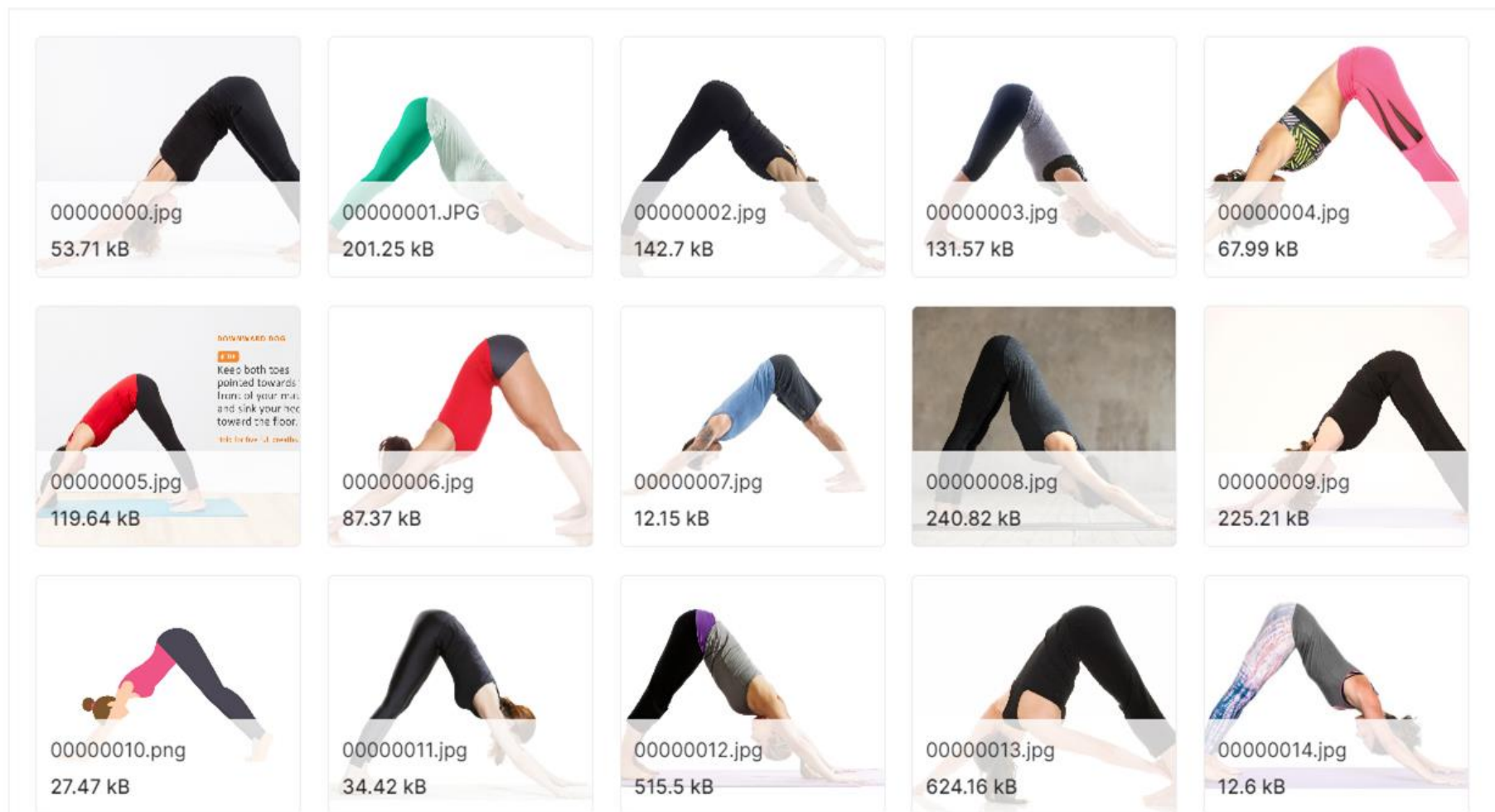
系統架構及技術



Technical & Structure

系統架構及技術

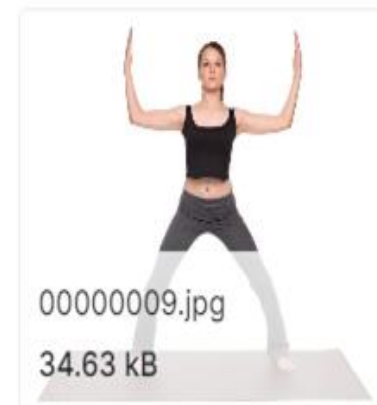
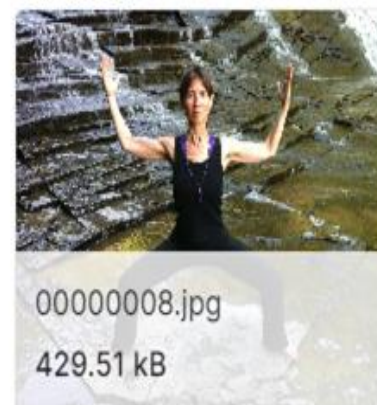
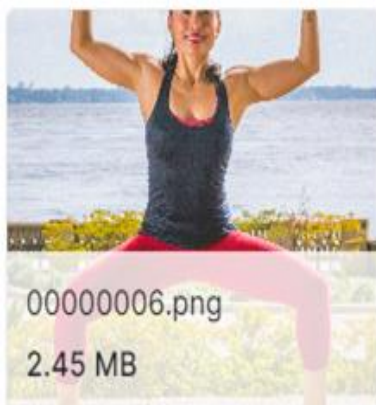
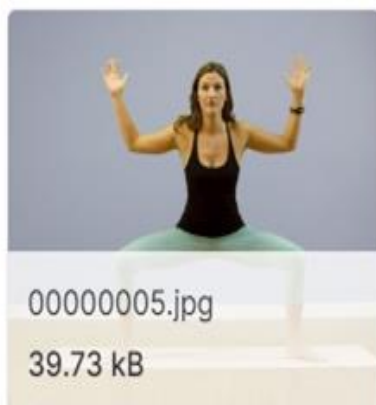
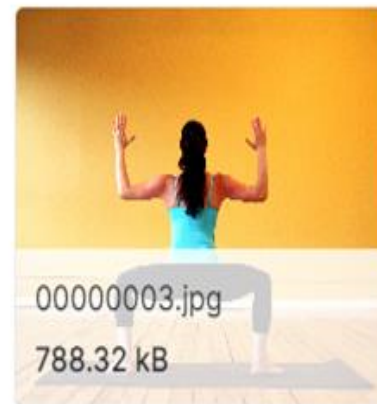
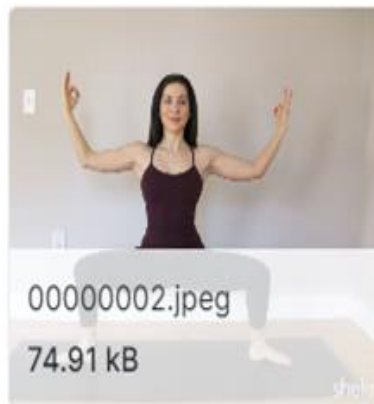
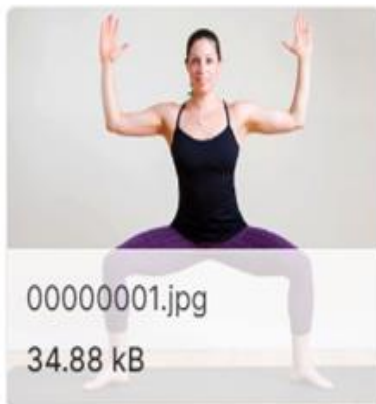
Downward dog 下犬式



Technical & Structure

系統架構及技術

Goddess 女神式

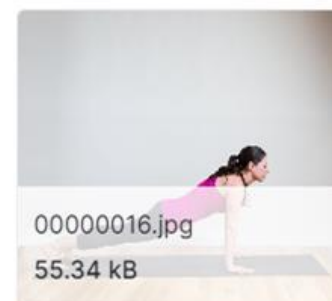
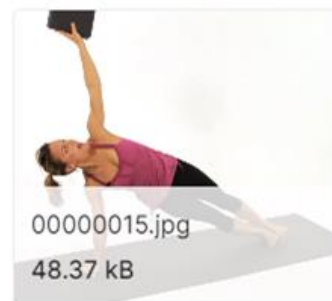
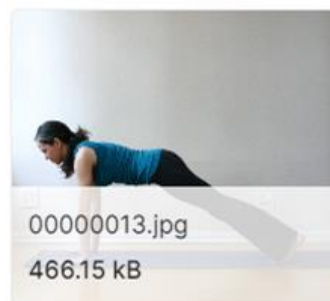
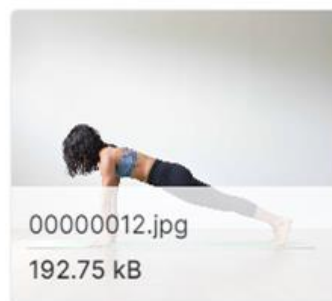
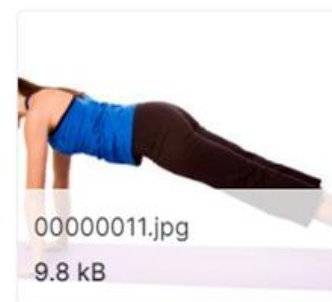
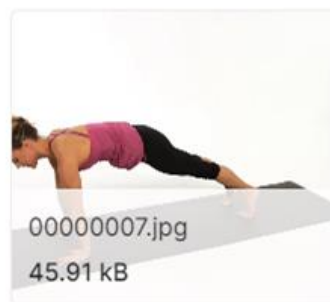
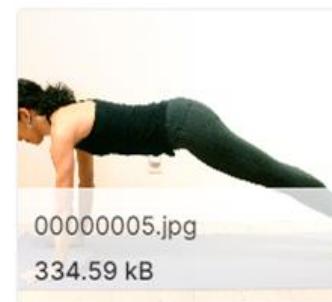
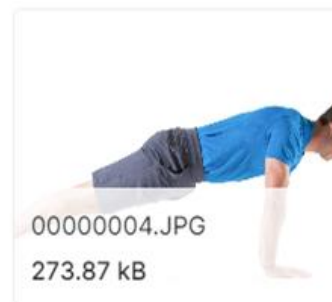


Technical & Structure

系統架構及技術

Plank

平板式

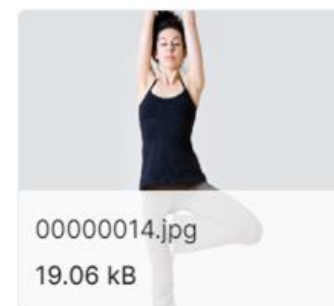
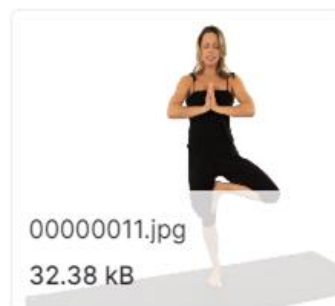
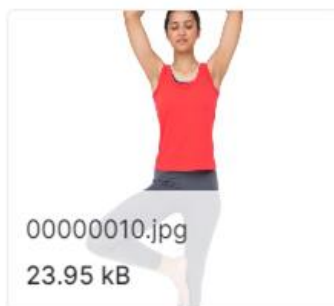
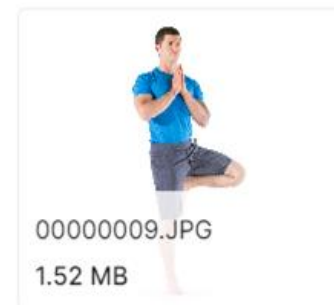
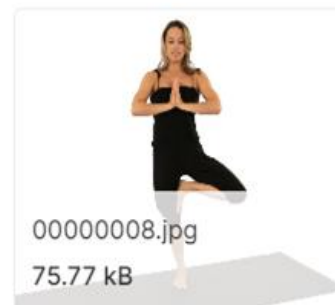
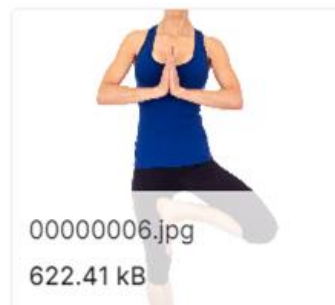
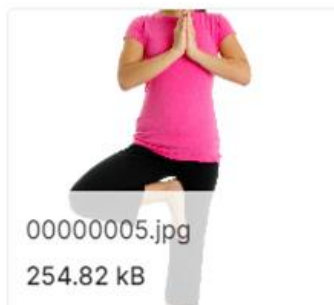
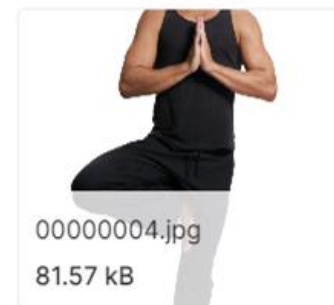
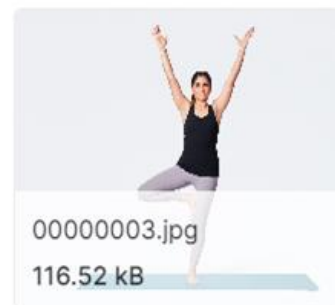
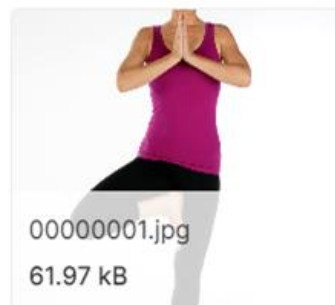
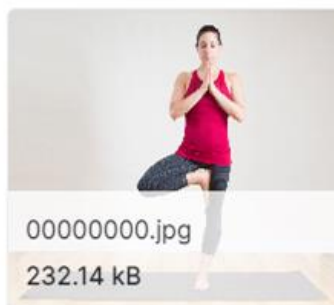


Technical & Structure

系統架構及技術

Tree

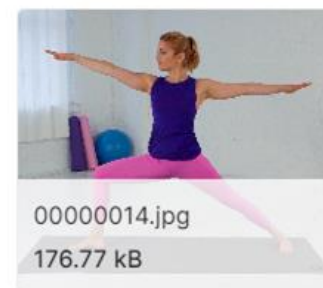
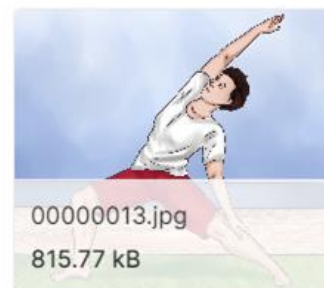
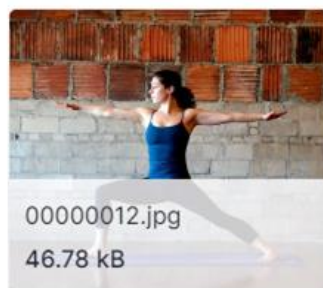
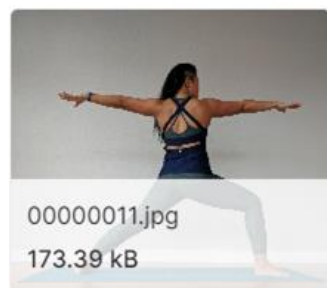
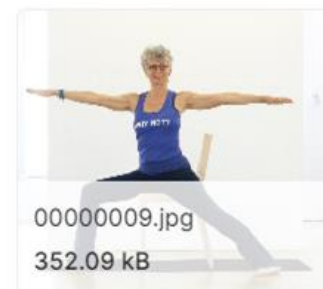
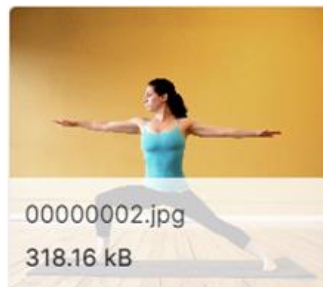
樹式



Technical & Structure

系統架構及技術

Warrior2 勇士二式



Technical & Structure

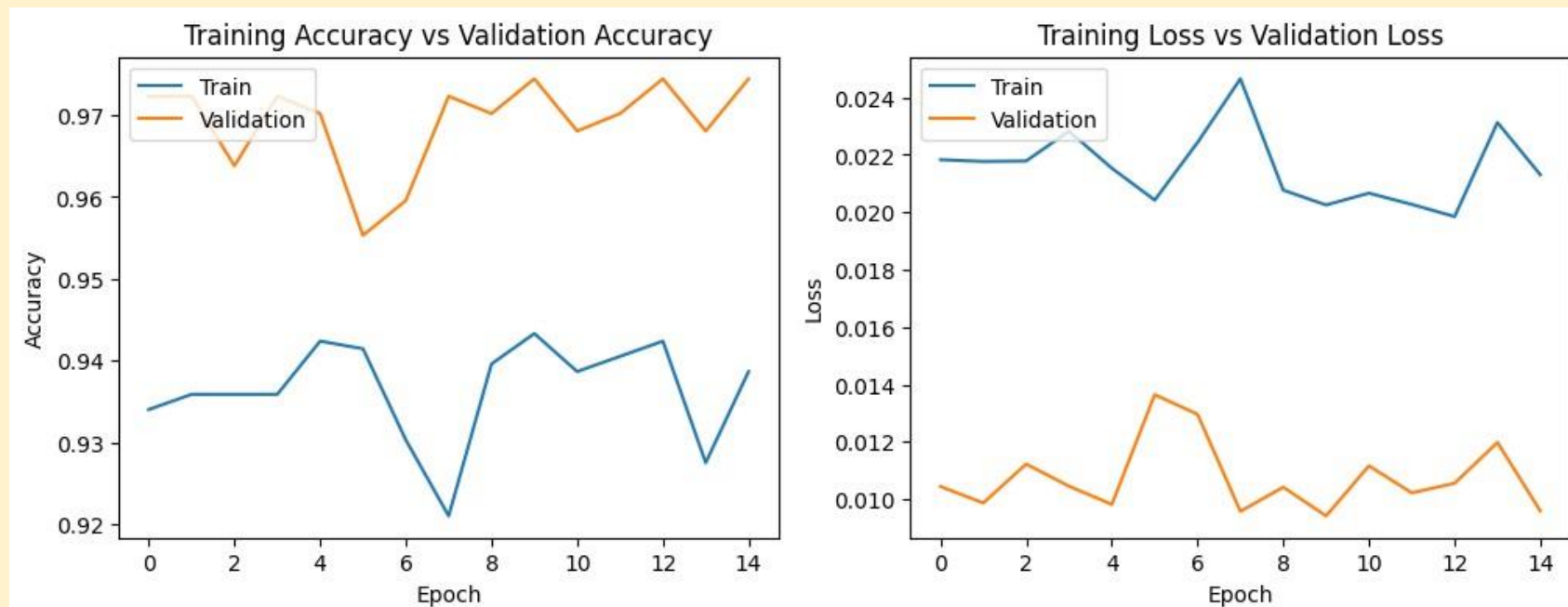
系統架構及技術

準確率

Accuracy
分類準確率
評估目標姿勢類型

MoveNet
瑜珈姿勢關鍵點
檢測

分類



final train accuracy = 94.70 , validation accuracy = 97.45

Technical & Structure

系統架構及技術

評估目標的瑜珈姿勢類型

Predicted Class: tree



Predicted Class: warrior2



Predicted Class: goddess



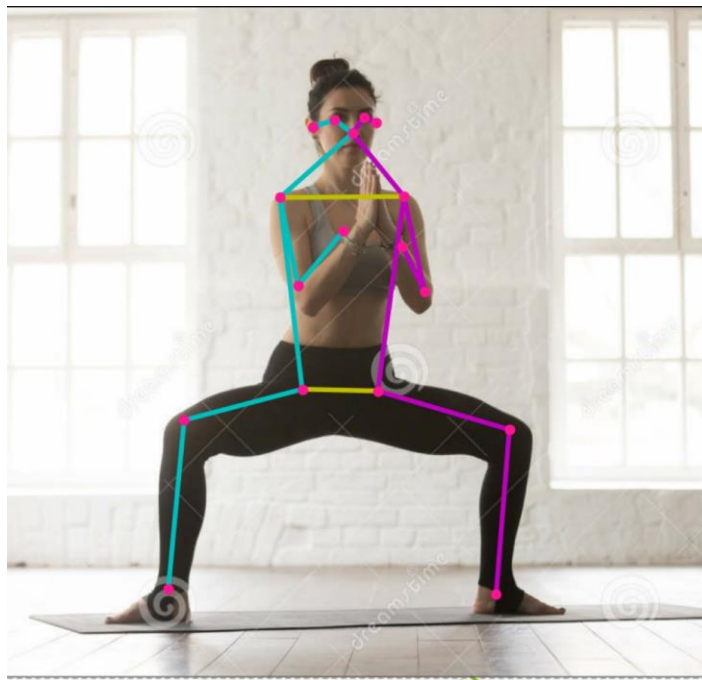
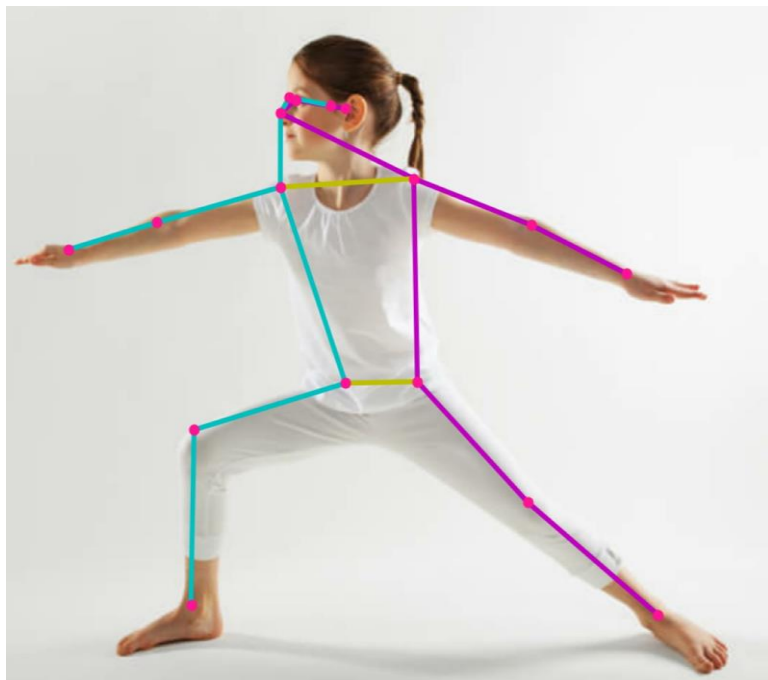
Predicted Class: downdog



Technical & Structure

系統架構及技術

瑜珈姿勢關鍵點檢測



Technical & Structure

系統架構及技術

人體動作判斷

篩選目標姿勢

透過使用者的臉部面向姿勢以及預測的姿勢種類來
配對合適的標準姿勢

檢測使用者面部方向

臉部面向為透過比較鼻子、左耳和右耳以及鼻子、左眼右眼的位置，例如：

- 鼻子在左耳和右耳的中間、以及鼻子在左眼或右眼下面時，為使用者面向正面
- 鼻子在左耳和右耳的左邊以及鼻子在左眼或右眼下面時，為使用者面向正面左邊
- 鼻子在左耳和右耳的右邊以及鼻子在左眼或右眼下面時，為使用者面向正面右邊
- 此外，判別使用者是否為背面時，可透過左右耳的位置是否與正面時相反，或透過左右耳到鼻子的距離來判定
- 鼻子在左眼和右眼的下面時，則使用者為倒立狀態，左右邊與上述情況相反

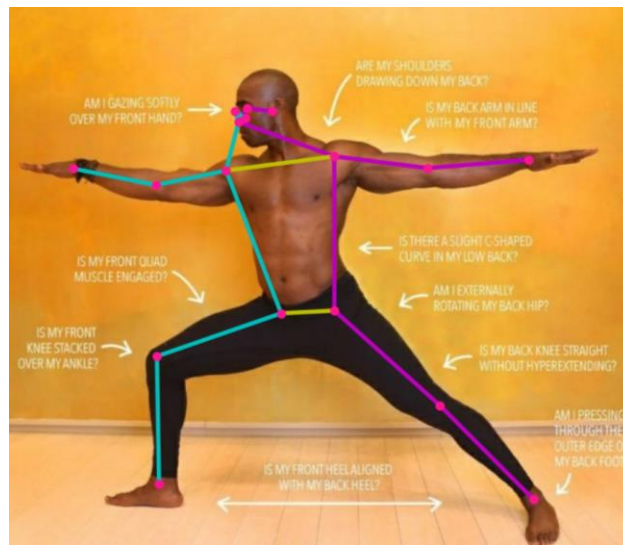
使用者面向正面



使用者面向背面轉左側



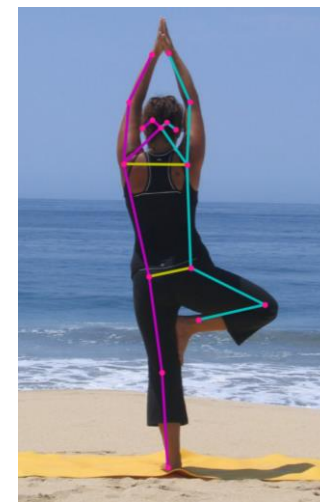
使用者面向正面轉右側



使用者面向正面轉左側



使用者面向背面



檢測使用者面部方向 [程式碼]

```
def face_position(target_keypoint):
    nose_y,nose_x,score=target_keypoint[0,0,0]
    left_eye_y,left_eye_x,score=target_keypoint[0,0,1]
    left_ear_y,left_ear_x,score=target_keypoint[0,0,3]
    right_ear_y,right_ear_x,score=target_keypoint[0,0,4]
    nose_to_left_ear = nose_x - left_ear_x
    nose_to_right_ear = nose_x - right_ear_x

    up=0
    if nose_y-left_eye_y<0:
        up=1
    str=""
    position=0
    if left_ear_x>nose_x and nose_x>right_ear_x:
        if up==0:
            str='使用者面向正面'
            position=1
        else:
            str='使用者倒立面向背面'
            position=2
    elif left_ear_x<nose_x and nose_x<right_ear_x:
        if up ==0:
            str='使用者面向背面'
            position=3
```

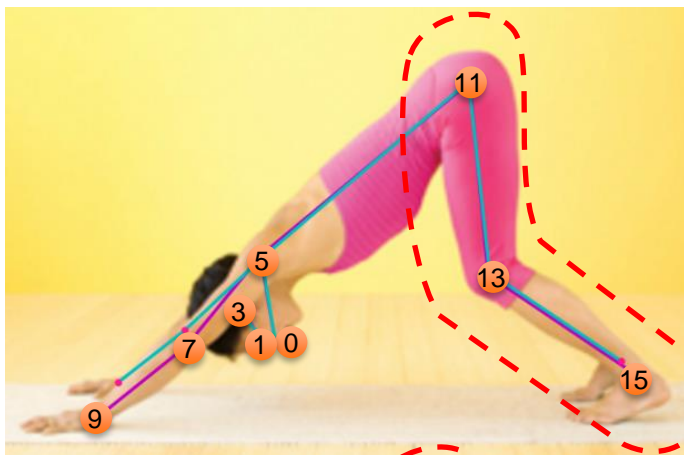
```
    else:
        str='使用者倒立面向正面'
        position=4
    elif left_ear_x>nose_x and nose_x<right_ear_x:
        if abs(nose_to_left_ear)-abs(nose_to_right_ear)<0:
            if up==0:
                str='使用者面向背面轉左側'
                position=5
            else:
                str='使用者倒立面向背面轉右側'
                position=6
        elif abs(nose_to_left_ear)-abs(nose_to_right_ear)>0:
            if up==0:
                str='使用者面向正面轉右側'
                position=7
            else:
                str='使用者倒立面向正面轉左側'
                position=8
        else:
            if up==0:
                str='使用者面向右側'
                position=9
            else:
                str='使用者倒立面向左側'
                position=10
```

```
    elif left_ear_x<nose_x and nose_x>right_ear_x:
        if abs(nose_to_left_ear)-abs(nose_to_right_ear)>0:
            if up==0:
                str='使用者面向背面轉右側'
                position=11
            else:
                str='使用者倒立面向背面轉左側'
                position=12
        elif abs(nose_to_left_ear)-abs(nose_to_right_ear)<0:
            if up==0:
                str='使用者面向正面轉左側'
                position=13
            else:
                str='使用者倒立面向正面轉右側'
                position=14
        else:
            if up ==0:
                str='使用者面向左側'
                position=15
            else:
                str='使用者倒立面向右側'
                position=16
    return position,str
```

Technical & Structure

系統架構及技術

目標對象姿勢與標準姿勢差異比較

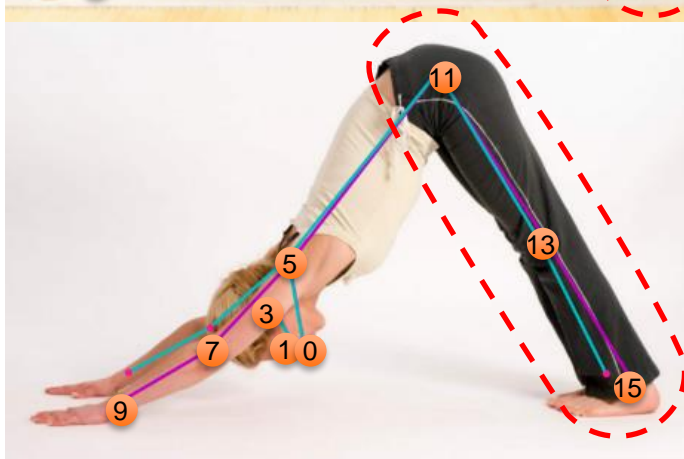


left side of body:

left_wrist to left_elbow
left_elbow to left_shoulder
left_ear to left_shoulder
left_shoulder to left_hip
left_hip to left_knee
left_knee to left_ankle

A 目標對象 左膝至左腳踝之角度為: 33.942974 度
B 標準姿勢 左膝至左腳踝之角度為: 58.406986 度
A - B = -24.464012 度

start_keypoint	end_keypoint	angle difference
9	7	-8.440210
7	5	-3.792137
3	5	2.797974
5	11	12.588238
11	13	27.397408
13	15	-24.464012



right side of body:

right_wrist to right_elbow
right_elbow to right_shoulder
right_ear to right_shoulder
right_shoulder to right_hip
right_hip to right_knee
right_knee to right_ankle

start_keypoint	end_keypoint	angle difference
10	8	-9.263792
8	6	-1.297108
4	6	7.727936
6	12	10.690731
12	14	24.915279
14	16	-29.538383

Technical & Structure

系統架構及技術

目標對象姿勢與標準姿勢差異比較

left side:

	start_keypoint	end_keypoint	angle difference
left_wrist to left_elbow	9	7	-65.143356
left_elbow to left_shoulder	7	5	-205.076050
left_ear to left_shoulder	3	5	-0.753609
left_shoulder to left_hip	5	11	-1.015976
left_hip to left_knee	11	13	3.092125
left_knee to left_ankle	13	15	0.942223

right side:

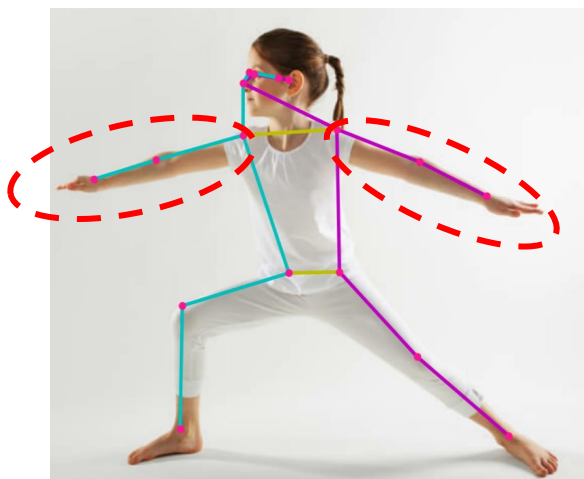
	start_keypoint	end_keypoint	angle difference
right_wrist to right_elbow	10	8	59.378387
right_elbow to right_shoulder	8	6	-140.871124
right_ear to right_shoulder	4	6	1.608948
right_shoulder to right_hip	6	12	-6.757362
right_hip to right_knee	12	14	0.949799
right_knee to right_ankle	14	16	-9.435909



Technical & Structure

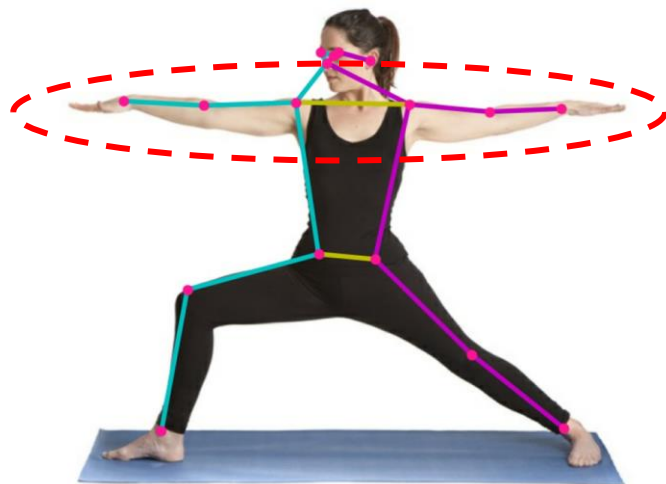
系統架構及技術

目標對象姿勢與標準姿勢差異比較



left side:

left_wrist to left_elbow
left_elbow to left_shoulder
left_ear to left_shoulder
left_shoulder to left_hip
left_hip to left_knee
left_knee to left_ankle



right side:

right_wrist to right_elbow
right_elbow to right_shoulder
right_ear to right_shoulder
right_shoulder to right_hip
right_hip to right_knee
right_knee to right_ankle

A 目標對象 左手腕至左手肘的角度為: -154.33853度
B 標準姿勢 左手腕到左手肘的角度為: 177.61024度
A - B = -331.9488 度

start_keypoint	end_keypoint	angle difference
9	7	-331.948792
7	5	15.192474
3	5	-4.412685
5	11	-13.519119
11	13	1.377876
13	15	0.572403

start_keypoint	end_keypoint	angle difference
10	8	-19.415665
8	6	-13.110435
4	6	-6.132851
6	12	-10.899956
12	14	-1.004547
14	16	-9.308151

Technical & Structure

系統架構及技術

目標對象姿勢與標準姿勢差異比較



left side:

left_wrist to left_elbow
left_elbow to left_shoulder
left_ear to left_shoulder
left_shoulder to left_hip
left_hip to left_knee
left_knee to left_ankle

start_keypoint	end_keypoint	angle difference
9	7	57.840782
7	5	20.713463
3	5	5.261414
5	11	0.688614
11	13	-12.239449
13	15	-3.059662

right side:

right_wrist to right_elbow
right_elbow to right_shoulder
right_ear to right_shoulder
right_shoulder to right_hip
right_hip to right_knee
right_knee to right_ankle

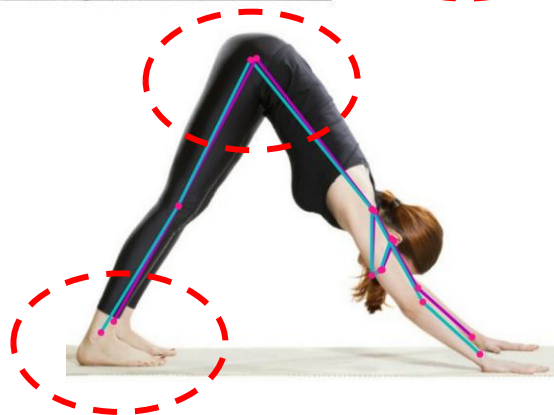
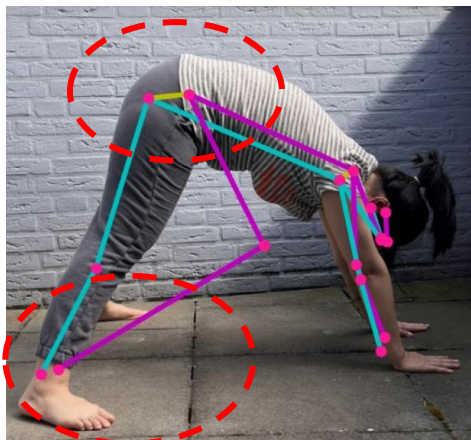
start_keypoint	end_keypoint	angle difference
10	8	-42.960648
8	6	-37.291786
4	6	1.007690
6	12	4.029533
12	14	17.498779
14	16	10.414352

A 目標對象 左手腕到左手肘的角度為: 64.01941度
B 標準姿勢 左手腕到左手肘的角度為: 6.1786265度
A - B = 57.840782 度

Technical & Structure

系統架構及技術

目標對象姿勢與標準姿勢差異比較



left side:

left_wrist to left_elbow
left_elbow to left_shoulder
left_ear to left_shoulder
left_shoulder to left_hip
left_hip to left_knee
left_knee to left_ankle

start_keypoint	end_keypoint	angle	difference
9	7		26.815895
7	5		27.640854
3	5		-4.336105
5	11		-27.296547
11	13		-54.277096
13	15		29.548500

right side:

right_wrist to right_elbow
right_elbow to right_shoulder
right_ear to right_shoulder
right_shoulder to right_hip
right_hip to right_knee
right_knee to right_ankle

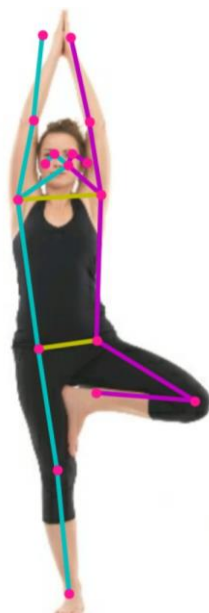
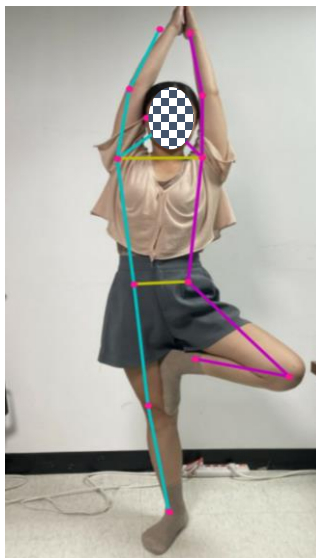
start_keypoint	end_keypoint	angle	difference
10	8		31.734306
8	6		16.934258
4	6		-15.564301
6	12		-27.836884
12	14		-9.325287
14	16		-4.154968

Technical & Structure

系統架構及技術

目標對象姿勢與標準姿勢差異比較

Predicted Class: tree



left side:

left_wrist to left_elbow
left_elbow to left_shoulder
left_ear to left_shoulder
left_shoulder to left_hip
left_hip to left_knee
left_knee to left_ankle

start_keypoint	end_keypoint	angle difference
9	7	1.797142
7	5	9.325912
3	5	2.235481
5	11	3.968666
11	13	14.776613
13	15	7.262375

right side:

right_wrist to right_elbow
right_elbow to right_shoulder
right_ear to right_shoulder
right_shoulder to right_hip
right_hip to right_knee
right_knee to right_ankle

start_keypoint	end_keypoint	angle difference
10	8	18.196579
8	6	-2.228691
4	6	-4.746422
6	12	1.045639
12	14	2.923149
14	16	-4.443985

— Optimization Evaluation

效能評估

混淆矩陣

Confusion Matrix

	precision	recall	f1-score	support
downdog	0.99	1.00	0.99	97
goddess	0.96	0.96	0.96	80
plank	0.97	0.97	0.97	115
tree	1.00	0.99	0.99	69
warrior2	0.95	0.96	0.96	109
accuracy			0.97	470
macro avg	0.98	0.98	0.98	470
weighted avg	0.97	0.97	0.97	470

‘Weighted’: 將計算加權平均精準度，即每個類別的精準度乘以其在數據集中的支持數量，然後將它們相加後除以總的樣本數量。這種方法對於樣本不平衡的情況同樣有用。

Confusion Matrix:

```
[[ 97  0  0  0  0]
 [  0 77  0  0  3]
 [  1  1 111  0  2]
 [  0  0  1 68  0]
 [  0  2  2  0 105]]
```

Accuracy: 0.9744680851063829

加權平均Precision: 0.9745305166757734

加權平均Recall: 0.9744680851063829

加權平均F1-score: 0.9744769756807199

— Problem Occured

遭遇問題

- 一開始有些訓練的檔案有混入png檔，利用OpenCV轉檔成JPG後，問題排除。
- 改寫程式，先使用MoveNet標關鍵點增加模型準確率。



Conclusion

結論

技術REVIEW

- STEP1 : 使用OpenCV資料前處理，使圖檔大小類型一致。
- STEP2 : 使用由google訓練好的MoveNet模型來資料前處理（速度快）。先透過關鍵點檢測模型把圖片的關鍵點座標都找出。
- STEP3 : 瑜珈姿勢預測（CNN關鍵點座標辨識），辨識目標照片屬於哪一類瑜珈姿勢。
- STEP4 : 以MoveNet關鍵點檢測，確認動作準確度。

優勢

CNN模型：對於非圖像數據，如果數據具有空間結構，CNN也可以從中受益。例如，某些非圖像數據可能包含網格狀結構或空間佈局，CNN可以有效地處理這些情況。

MoveNet：可以檢測人體的17個關鍵點。模型輕量化，能減少運算負擔，並加快姿態檢測的速度。

Yoga Master Overpower：高效、快速、準確。即時執行瑜珈姿態檢測，清爽無負擔。

感想

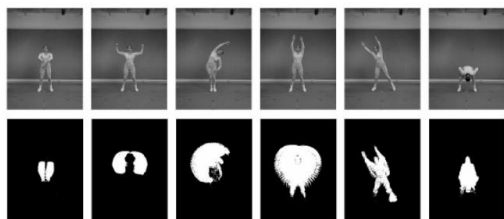
- 解決問題，要先問對問題！
- 如果能夠讓使用者更方便、即時將影像資訊載入模型，並且使用chatbot自動回覆給予使用者建議，可用於瑜珈相關的商業應用。但因礙於有限的時間及資源，所以我們決定專注在訂定的主題。

Industry Case Sharing

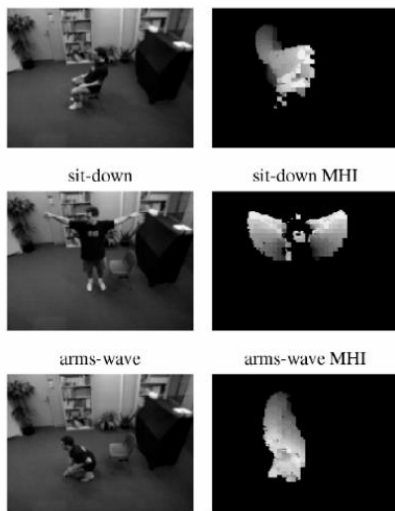
產學界案例分享

視覺式體操動作辨識系統

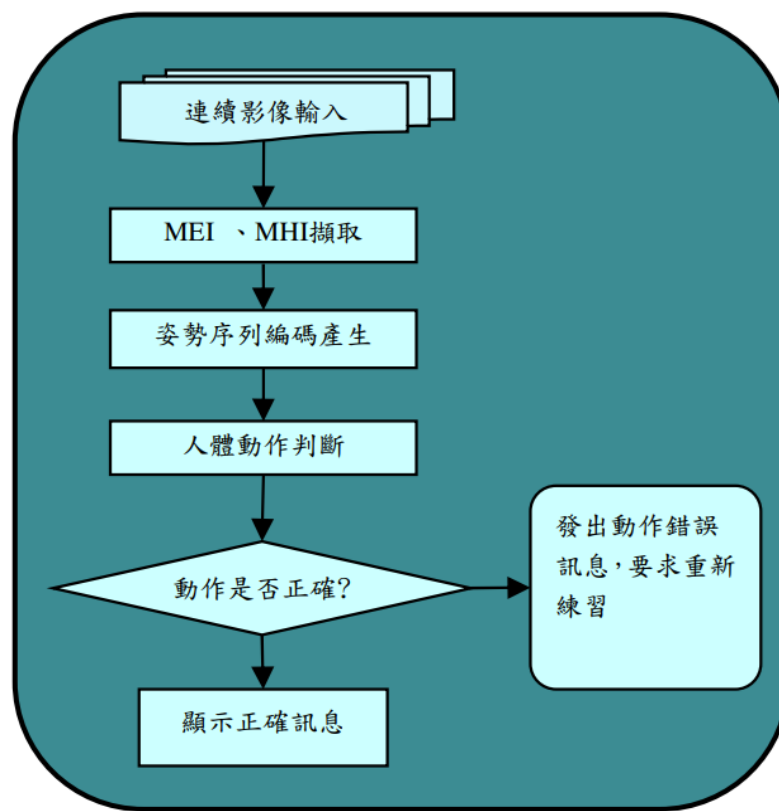
Vision-based Gymnastics Motion Recognition System



圖六、六種不同體操動作所對應之MEI圖[13]



圖七 三種不同動作所對應之 MHI 圖[13]



本計畫設計一套以電腦視覺為基礎偵測人體全身肢體動作之系統，將連續之體操運動影像輸入電腦後做肢體動作判斷辨識其動作種類或發出錯誤動作訊息。

- 以motion-energy images (MEI) motion-history images(MHI) 做為比對的依據

Industry Case Sharing

產學界案例分享

Johnson@Mirror 新概念健身魔鏡



Industry Case Sharing

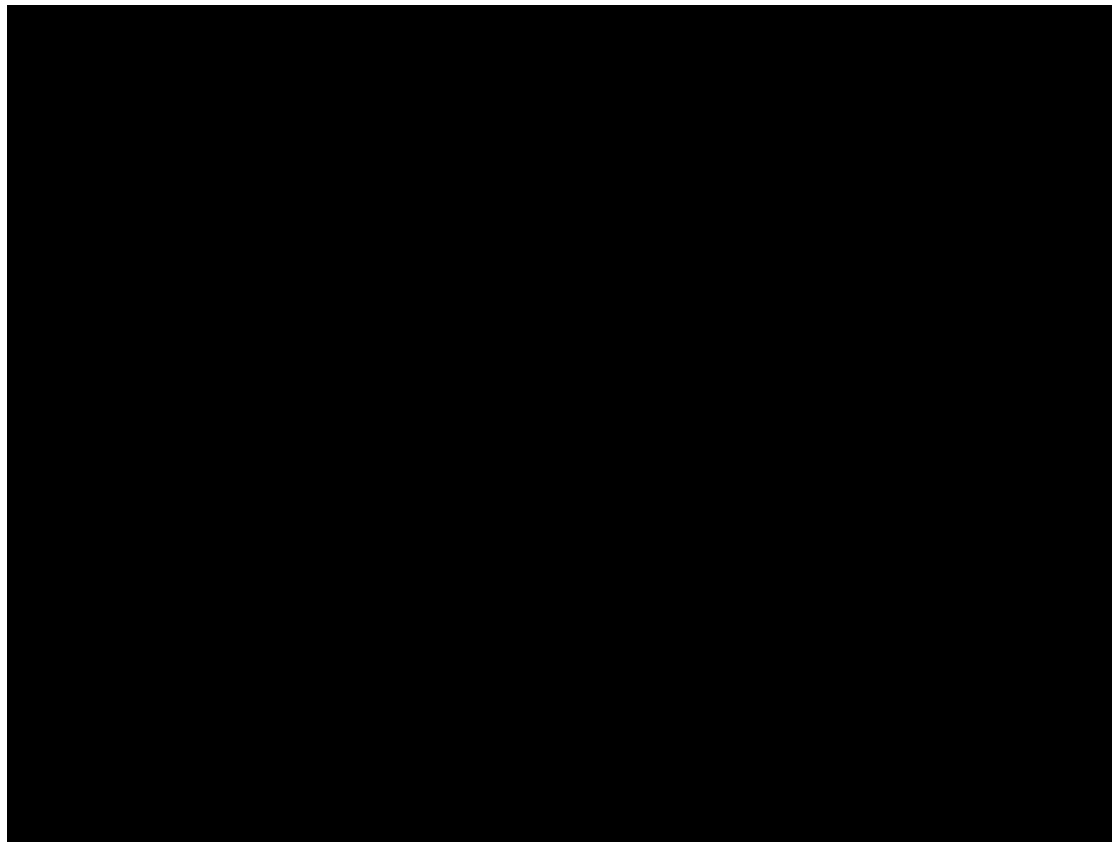
產學界案例分享

Hong Kong Univisual Intelligent Technology (HKUIT)

透過物體偵測、物體追蹤及人體姿勢分析，自動識別人體13個關節節點並作出分析。公司開發了兩項產品，分別是**人工智能游泳安全及泳姿分析系統**，以及**人體姿勢分析開發套件(SDK)**，前者應用於室內泳池內，後者則供開發商設計不同運動分析方案如瑜珈應用程式。

在泳池內不同位置包括水底、天花、池邊等安裝多個網絡攝影機，以及以太網絡(Ethernet)，收集及分析泳客泳姿影像數據，例如游泳頻率、幅度、撥水速度、手腳協調度等，救生員透過平板電腦辨識遇溺者，游泳教練則可協助泳手改善技術。

<https://www.hkuit.com/>



分工表

Organization Chart

組員姓名/ 工作類別	組長: 047 劉建宏	組員: 014 陳亭均	組員: 015 梁毓琳	組員: 025 毛彥文	組員: 051 于復申	組員: 046 邵文綺
主題綱要	V					
程式修正	V		V			
文字編排	V	V		V	V	V
資料尋找/提供	V	V	V	V	V	V
上台發表	V	V	V	V		V

Reference

參考文獻

- 喬山健康科技 (n.d) 。 Johnson@Mirror 新概念健身魔鏡 。 *Johnson@Mirror 新概念健身魔鏡- JOHNSON* 喬山健康科技-跑步機、按摩椅。 <https://www.johnsonfitness.com.tw/prod/?q=MIRROR>
- 顏羽君 (2008年) 。 視覺式體操動作辨識系統 *Vision-based Gymnastics Motion Recognition System* 。 國立臺灣師範大學。
- HKUIT 。 (2023, 5月10日) 。 【人工智能】港大研AI偵測技術 分析人體動作實時警報溺水跌倒等意外。 TOPick 。 <https://www.it-square.hk/archives/15362/ai%E5%88%86%E6%9E%90%E8%82%A2%E9%AB%94%E5%8B%95%E4%BD%9C-%E6%B8%9B%E4%BD%8E%E9%81%87%E6%BA%BA%E6%94%B9%E5%96%84%E9%81%8B%E5%8B%95%E5%A7%BF%E5%8B%A2/>
- Suradech Kongkiatpaiboon, Burq Latif.(2021, November 23). 03_keypoint_movenet_v2. <https://www.kaggle.com/code/suradechk/03-keypoint-movenet-v2>
- Aayush Mishra. (2021, June 03). Yoga Pose Detection. <https://www.kaggle.com/code/aayushmishra1512/yoga-pose-detection/notebook>
- Ronny Votel, Na Li. (2021, May 17). Next-Generation Pose Detection with MoveNet and TensorFlow.js. <https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html>
- Aayush Mishra. (2021, June 03). YogaNet - VGG19 <https://www.kaggle.com/code/aayushmishra1512/yoganet-vgg19>
- VK. (2022, Feb 21). Pose_Prediction|Generate_CSV_Keypoints|MediaPipe <https://www.kaggle.com/code/venatkumar001/pose-prediction-generate-csv-keypoints-mediapipe>
- OXXO.STUDIO.(2023, Aug 01). Mediapipe 姿勢偵測 (Pose). <https://steam.oxxostudio.tw/category/python/ai/ai-mediapipe-pose.html>
- Yoga Poses Dataset. (n.d.). Kaggle.Com.. <https://www.kaggle.com/datasets/niharika41298/yoga-poses-dataset>