**PLA Matching with watershed API**

**包含模組:**

* main.py (使用所有模組的最上層檔案)
* find\_roi.py
* skeletonize\_bound.py
* model\_scale.py
* matching.py
* muscle\_bound.py
* muscle\_marker.py
* chamber\_marker.py
* mark\_position.py

**簡介:**

Parasternal Long Axis 的腔室及肌肉範圍，利用與標準模型匹配(Matching)的做法，去初步找出肌肉的區域，接著利用Watershed 找出範圍。

**使用方式:**

說明除了main.py的其他模組使用方式，程式碼的Example則是main.py內容。Matching 順序 A > C > B

* **Module Name: find\_roi.py**

此模組使用 Python Class 來創建，Class 名稱為 FindROI，以下為物件創建必要的輸入參數

|  |  |  |
| --- | --- | --- |
| 參數名稱 | 資料類型 | 說明 |
| video\_path | str | 輸入影片路徑 |

* **Method:**

1. **get\_frame(self): 讀取每幀影像**

**return:** 當前frame，numpy.ndarray

(這裡沒有使用到)

1. **roi\_region(self, input\_path[, mask\_threshold=10[, kernel\_size=(3,3)[, morph\_iter=3]]]): 找出超音波影像(ROI)區域**

**parameters:**

input\_path: 輸入影片路徑，str

mask\_threshold: mask的門檻值，默認10，int

kernel\_size: 形態學的kernel大小，默認(3, 3)，tuple(int, int)

morph\_iter: 形態學迭代次數，默認3，int

**return:** None

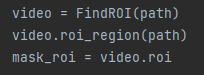
**attributes:**

self.roi: ROI 的mask 區域，numpy.ndarray，二值圖

self.ox, self.oy: ROI 的扇形圓心座標，int

self.radius: ROI扇形半徑，int

Example:



Video 為 FindROI 類別實例，給path輸入路徑後，呼叫roi\_region(path) 方法，透過讀取屬性roi，取得mask。

* **Module Name: skeletonize\_bound.py**
* **Function:**

1. **skeleton\_bound(file\_name, roi): 利用骨架圖的範圍調整ROI邊界大小**

**parameters:**

file\_name: 骨架圖的檔案路徑，str

roi: mask區域，numpy.ndarray，二值圖(可以使用find\_roi裡的roi屬性值)

**return:** 回傳帶有5個元素的列表

top: 上邊界，int

bottom: 下邊界，int

left: 左邊界，int

right: 右邊界，int

radius: 半徑，int

Example:



skeleton\_info: 變數名

直接將模組import 進來後，直接呼叫function並給予對應參數即可使用。

* **Module Name: model\_scale.py**
* **Function:**

1. **adjust\_scale(model\_all, model\_A, model\_B, model\_C, radius):**

**調整標準模型比例**

**parameters:**

model\_all: 完整的標準模型，numpy.ndarray，模型圖片

model\_A: 模型的 A 區段，numpy.ndarray，模型A區段圖片

model\_B: 模型的 B 區段，numpy.ndarray，模型B區段圖片

model\_C: 模型的 C 區段，numpy.ndarray，模型C區段圖片

radius: 骨架圖的有效半徑(可以使用skeletonize\_bound 回傳值)，int

**return:**

output\_model\_A: resize 後的 model\_A，numpy.ndarray，模型A區 段圖片

output\_model\_B: resize 後的 model\_B，numpy.ndarray，模型B區 段圖片

output\_model\_C: resize 後的 model\_C，numpy.ndarray，模型A區 段圖片

Example:



* **Module Name: matching.py**
* **Function:**

1. **match\_muscleA(frame, A[, v\_range=range(15, 80, 5)[, h\_range=range(-30, 30, 6)[, t\_range=(-10, 20, 5)]]]): 擬合模型A區段**

**parameters:**

frame: 每幀影像，numpy.ndarray，3通道

A: 模型A區段，numpy.ndarray(可以使用adjust\_scale回傳值)

v\_range: 垂直範圍(start, end, step)，默認range(15, 80, 5)，iterator

h\_range: 水平範圍(start, end, step)，默認range(-30, 30, 6)，iterator

t\_range: 旋轉範圍(start, end, step)，默認range(-10, 20, 5)，iterator

**return:**

model\_best\_A: 模型的最佳擬合結果，numpy.ndarray(模型圖)

best\_fitting\_A: 模型和影像的最佳擬合結果，numpy.ndarray(影像)

Example:



1. **match\_muscleB(frame, B, seg\_B[, v\_range=range(-10, 10, 5)[, h\_range=range(-50, 10, 5)[, t\_range=(-10, 10, 2)]]]): 擬合模型B區段**

**parameters:**

frame: 每幀影像，numpy.ndarray，3通道

B: 模型B區段，numpy.ndarray(可以使用adjust\_scale回傳值)

seg\_B: 消除已經擬合的(A, C)區段，降低重複擬合的可能性， numpy.ndarray

v\_range: 垂直範圍(start, end, step)，默認range(-10, 10, 5)，iterator

h\_range: 水平範圍(start, end, step)，默認range(-50, 10, 5)，iterator

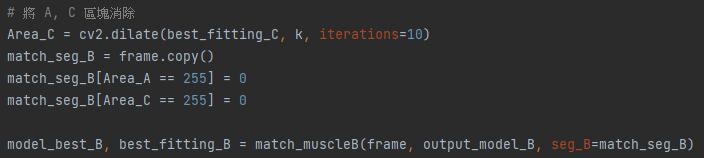
t\_range: 旋轉範圍(start, end, step)，默認range(-10, 10, 2)，iterator

**return:**

model\_best\_B: 模型的最佳擬合結果，numpy.ndarray(模型圖)

best\_fitting\_B: 模型和影像的最佳擬合結果，numpy.ndarray(影像)

Example:



消除已經matching過的區域，這裡是利用形態學的做法，對原始影像進行消除(方法並非唯一)

1. **match\_muscleC(frame, C, seg\_C[, v\_range=range(-32, 0, 4)[, h\_range=range(-20, 6, 4)[, t\_range=(-6, 10, 2)]]]): 擬合模型C區段**

**parameters:**

frame: 每幀影像，numpy.ndarray，3通道

C: 模型C區段，numpy.ndarray(可以使用adjust\_scale回傳值)

seg\_C: 消除已經擬合的(A)區段，降低重複擬合的可能性， numpy.ndarray

v\_range: 垂直範圍(start, end, step)，默認range(-32, 0, 4)，iterator

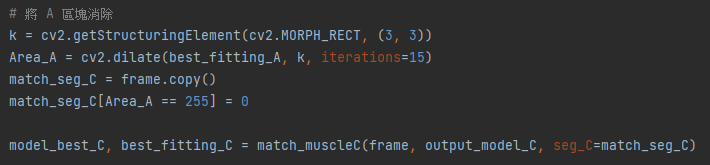
h\_range: 水平範圍(start, end, step)，默認range(-20, 6, 4)，iterator

t\_range: 旋轉範圍(start, end, step)，默認range(-6, 10, 2)，iterator

**return:**

model\_best\_C: 模型的最佳擬合結果，numpy.ndarray(模型圖)

best\_fitting\_C: 模型和影像的最佳擬合結果，numpy.ndarray(影像)



* **Module Name: muscle\_bound.py**
* **Function:**

1. **muscle\_bounding(src[, min\_thres=50[, erode\_iterations=10]]): 找出肌肉的邊界**

**parameters:**

src: matching後合併的圖像，numpy.ndarray，灰階圖(Example 會說明)

min\_thres: 二值化的最小門檻值，默認50，int

erode\_iterations: 形態學侵蝕次數，默認10，int

**return:**

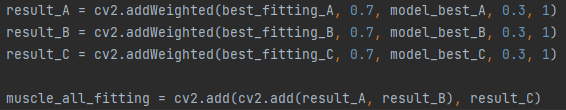
gray\_bound\_region: 未經過侵蝕後的肌肉邊界mask， numpy.ndarray，二值圖

erode\_bound\_region: 侵蝕後的肌肉邊界mask，numpy.ndarray，二 值圖

Example:



muscle\_all\_fitting為 matching 後的結果相加得到的圖像，如下程式碼



* **Module Name: muscle\_marker.py**
* **Function:**

1. **handle\_muscle(muscles, muscleA, muscleB, muscleC[, min\_thres=50]): 將每個肌肉區段的破碎區塊連起來，且搜尋瓣膜可能的位置**

**parameters:**

muscles: matching後的影像相加結果，numpy.ndarray，三通道

muscleA: matching後的A區段結果(best\_fitting\_A)

muscleB: matching後的B區段結果(best\_fitting\_B)

muscleC: matching後的C區段結果(best\_fitting\_C)

min\_thres: 二值化的最小門檻值，默認50，int

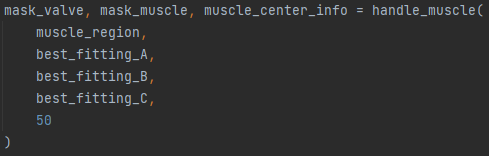
**return:**

mask\_valve: 利用handle\_valve()方法搜尋瓣膜可能的位置， numpy.ndarray，二值圖

mask\_muscle: 三個區段的肌肉mask，numpy.ndarray，二值圖

muscle\_center\_info: 儲存三段肌肉的輪廓中心點，list(list(int, int))

Example:



muscle\_region的內容為下圖所示



1. **handle\_valve(cntA, cntB, cntC): 搜尋二尖瓣及動脈瓣膜位置**

**parametets:**

cntA: 肌肉A區段的輪廓，list(contours)

cntB: 肌肉B區段的輪廓，list(contours)

cntC: 肌肉C區段的輪廓，list(contours)

**return:**

mv\_ax, mv\_ay: 二尖瓣位置第一點座標，tuple(int, int)

mv\_bx, mv\_by: 二尖瓣位置第二點座標，tuple(int, int)

av\_cx, av\_cy: 動脈瓣膜位置第一點座標，tuple(int, int)

av\_bx, av\_by: 動脈瓣膜位置第二點座標，tuple(int, int)

此function已經由handle\_muscle()呼叫，因此只要呼叫handle\_muscle()即可，不需額外呼叫handle\_valve()

* **Module Name: chamber\_marker.py**
* **Function:**

1. **handle\_chamber(gray\_frame, bound, center\_info[, min\_thres=195]): 找出腔室的marker**

**parameters:**

gray\_frame: 灰階圖像，numpy.ndarray

bound: 肌肉的邊界(腔室會被肌肉包圍著)，可以使用 muscle\_bounding的回傳值erode\_bound\_region， numpy.ndarray

center\_info: 肌肉中心點的座標(可以使用handle\_muscle()的回傳結 果，muscle\_center\_info)，list(list(int, int))

min\_thres: 二值化的最小門檻值，默認195，int

**return:**

mask\_filter: 腔室的marker，numpy.ndarray，二值圖

center\_info: 腔室輪廓的中心點座標資訊，list(list(int, int))

Example:



此處的gray\_frame，有做過mask處理。把在肌肉邊界外和瓣膜的 區塊mask掉(erode\_bound\_region及mask\_valve)。

* **Module Name: mark\_postion.py**
* **Function:**

1. **handle\_watershed(frame, bound, mark, center\_info):**

**對腔室和肌肉做 watershed**

**paremeters:**

frame: 原始影像，numpy.ndarray，3通道

bound: 肌肉的邊界，(可以使用muscle\_bounding的回傳值 ，gray\_bound\_region，numpy.ndarray)

mark: 要做watershed 的marker(可以是肌肉、腔室或肌肉+腔 室)，numpy.ndarray，二值圖

center\_info: 腔室輪廓中心點座標(可以使用handle\_chamber()的回 傳值，center\_info)，list(list(int, int))

**return:**

curr\_result: 經過watershed 後的影像，numpy.ndarray，3通道

Example:



此處的ori\_bound為gray\_bound\_region，mask\_filter為腔室+肌肉 的marker。