**A4C LV Muscle Semantic API**

**包含模組:**

* Main.py (使用所有模組的最上層檔案)
* PrecessesCtrl.py
* FileIO.py
* DCMToAVI.py
* Preprocessing.py
* A4CSegmentation.py
* MultiThreshold.py (請參照 **MultiThreshold API**)
* MuscleSampling.py
* A4CGLS.py

**簡介:**

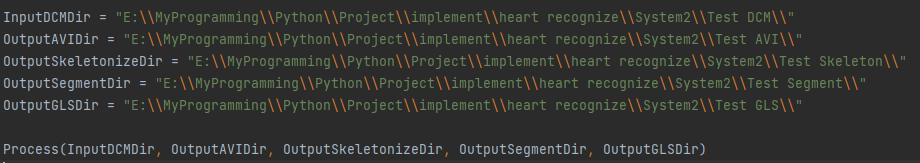
由於此資料夾放置的是A4C View到Muscle Semantic的系統化架構，因此只需要對應好輸入輸出及可以使用。此外在Class底下有用前底線命名的方法皆不需要直接呼叫。(Kmeans的部分詳細內容會寫在教學文件，這裡只提輸入輸出)

**使用方式:**

* **Module Name: Main.py**

僅需要把資料夾的路徑寫好，呼叫ProcessCtrl裡面的Process執行即可使用。

Example:



即使資料夾的路徑不存在，也可以被創建，因此可以輸入個人想要的位置。

* **Module Name: PrecessedCtrl.py**

控制整個流程的進行，如果新增東西即可直接在該檔案底下新增。

以下為輸入參數

|  |  |  |
| --- | --- | --- |
| 參數名稱 | 資料類型 | 說明 |
| InputDCMDir | str | 輸入DCM檔案資料夾路徑 |
| OutputAVIDir | str | 輸出AVI檔案資料夾路徑 |
| OutputSkeletonizeDir | str | 輸出骨架圖檔案資料夾路徑 |
| OutputSegmentDir | str | 輸出Segment影片檔案資料夾路徑 |
| OutputGLSDir | str | 輸出GLS影片檔案資料夾路徑 |

* **Module Name: FileIO.py**

負責處理讀取資料夾特定副檔名的檔案以及輸出計算結果後影片(此資料夾的任何檔案輸出皆由此模組處理)

* **Method:**

**1. AllFiles(DirPath[, extension\_name=’avi’]): 讀取DirPath資料夾 底下的所有符合extension\_name的檔案**

**parameters:**

DirPath:目標資料夾路徑，str

extension\_name: 副檔名的名稱，例如: mp4、avi、dcm等。不區 分大小寫，檔名前不需要加點(’.’)也可以使用。默認avi，str

**return:**

result: 所有符合條件檔案的絕對路徑列表，list(str)

Example:



搜尋所有在 InputDCMDir底下副檔名為dcm的檔案

**2. write\_video(FrameList, OutputPath[, fps=30]): 輸出影片到指定 的資料夾路徑**

**parameters:**

FrameList: 儲存每幀影像(一定要三通道)的列表， list(numpy.ndarray)

OutputPath: 影片輸出資料夾的路徑，str

fps: frames per second，每秒幾個畫面，默認30，int。

**return:** None

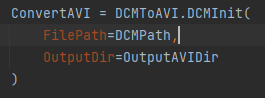
* **Module Name: DCMToAVI.py**

此模組使用Python Class 來創建，Class 名稱為DCMInit，以下為基本的輸入參數和存放avi路徑的屬性

|  |  |  |
| --- | --- | --- |
| 參數名稱 | 資料類型 | 說明 |
| FilePath | str | 輸入DCM檔案資料夾路徑 |
| OutputDir | str | 輸出AVI檔案資料夾路徑 |

|  |  |  |
| --- | --- | --- |
| 屬性名稱 | 資料類型 | 說明 |
| self.DCMFilePath | str | 存放輸入DCM檔案資料夾路徑 |
| self.OutputDir | str | 存放輸出AVI檔案資料夾路徑 |
| self.OutputDirPath | list(str) | 存放輸出AVI檔案資料夾路徑底下的所有檔案，可以避免已經轉換過的檔案重新轉換 |
| self.AVIPath | list(str) | 存放AVI檔案的路徑列表 |
| self.\_ConvDCMToAVI() | method | 將DCM檔案轉成AVI檔案的method，不需要直接呼叫。在創建class的物件實例時就會執行此方法了 |

Example:



* **Module Name: Preprocessing.py**

此模組用來處理影像預處理，例如: 骨架化、ROI

* **Function:**

**1. ROI(Path): 找出超音波影像(ROI)區域**

**parameters:**

Path: 影片的輸入路徑，str

**return:**

roi: ROI 的mask 區域，numpy.ndarray，二值圖

(ox, oy): ROI 的扇形圓心座標，int

radius: ROI扇形半徑，int

Example:



**2. Skeletonize(Path, OutputSkeletonDir): 將影片做骨架化後產生的 骨架圖輸出到指定資料夾路徑**

**parameters:**

Path: 影片的輸入路徑，str

OutputSkeletonDir: 輸出骨架圖檔案資料夾路徑

**return:** None

Example:

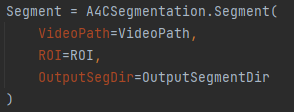
* **Module Name: A4CSegmentation.py**

此模組處理了A4C的腔室語意分析及定義二尖瓣位置。使用Python Class創建，Class名稱為Segment，以下為Segment必要的輸入參數及屬性

|  |  |  |
| --- | --- | --- |
| 參數名稱 | 資料類型 | 說明 |
| VideoPath | str | 輸入AVI影片路徑 |
| ROI | list | 使用**Preprocessing的ROI function回傳值當成輸入**，否則會出bug |
| OutputSegDir | str | 輸出Segment影片檔案資料夾路徑 |

|  |  |  |
| --- | --- | --- |
| 屬性名稱 | 資料類型 | 說明 |
| self.VideoPath | str | 存放輸入AVI影片路徑 |
| self.roi | numpy.ndarray | 超音波影像(ROI)區域 |
| self.OutputSegDir | str | 存放Segment影片檔案資料夾路徑 |
| self.ChamberCenX | int | 腔室中心點x座標 |
| self.ChamberCenY | int | 腔室中心點y座標 |
| self.MaskChamberBound | numpy.ndarray | 腔室範圍的mask |
| self.Centroids | dict | 存放四個腔室的(kmeans)質心點 |
| self.HistoryCenters | dict | 存放四個腔室的每幀中心點 |
| self.LeftPivotList | list(tuple(int,int)) | 存放左側二尖瓣支點的座標列表 |
| self.RightPivotList | list(tuple(int,int)) | 存放右側二尖瓣支點的座標列表 |
| self.\_CreateFeatures() | method | 創建Kmeans需要的特徵 |
| self.\_KmeansCluster(ClusterData) | method | 利用Kmeans分出四個腔室質心點 |
| self.\_KmeansAnomalyDetection(  ClusterFeature,  CentroidList,  OriginalData  firstPred  ) | method | 檢測Kmeans的四個質心點位置是否符合醫學上A4C的腔室位置 |
| self.\_ReKmenas(data) | method | 重新用Kmeans預測質心點 |
| self.\_FrameCenterAnomalyDetection(  CurrentCenters) | method | 檢測每幀影像中是否都有存在4個腔室，針對腔室點錯誤的情況處理 |

Example:



* **Method:**

1. **HandleHeartBound(self, Skeleton): 利用骨架圖找出心臟範圍**

**parameter:**

Skeleton: 輸入3通道骨架圖，numpy.ndarray

**return:** None

Example:



1. **Semantic\_FindValve(self[, isOutputSegVideo=False]): 腔室語意分析以及定義二尖瓣支點的位置**

**parameter:**

isOutputSegVideo: 是否要輸出Segmentation的影片，默認False，bool

**return:** None

Example:



**3. \_CreateFeatures(self): 創建Kmeans需要的特徵**

**parameter:** None

**return:**

CenterData: Kmeans 的特徵矩陣，numpy.ndarray

**4. \_KmeansCluster(ClusterData): 利用Kmeans分出四個腔室質心點**

**parameter:**

ClusterData: 輸入 \_CreateFeatures() 的回傳值

**return:**

self.\_KmeansAnomalyDetection(

ClusterFeature=Cluster\_Centers,

CentroidList=Centroid\_pts,

OriginalData=ClusterData,

firstPred=YPred

)

變數說明:

Cluster\_Centers: Kmeans的中心點特徵矩陣

Centroid\_pts: 四個質心點座標列表

ClusterData: 聚類的特徵矩陣資料

YPred: 第一次Kmeans預測結果

5. **\_KmeansAnomalyDetection(ClusterFeature, CentroidList, OriginalData, firstPred): 檢測Kmeans的四個質心點位置是否符合 醫學上A4C的腔室位置**

**parameters:**

ClusterFeature: Kmeans的中心點特徵矩陣

CentroidList: 四個質心點座標列表

OriginalData: 聚類的特徵矩陣資料

firstPred: 第一次Kmeans預測結果

**return:**

adjust\_Centroid: 調整過後的四個職心點座標，dict

**6. \_ReKmeans(data): 若質心座標異常，則在** **KmeansAnomalyDetection() 方法裡會呼叫此函數，重新進行預測**

**parameter:**

data: 新的特徵矩陣資料，透過KmeansAnomalyDetection() 調整

**return:**

Centroid\_pts2: 新的質心位置列表，list(tuple(int,int))

ypred: 新的分類預測標籤

**7. \_FrameCenterAnomalyDetection(CurrentCenters): 檢測每幀影像中 是否都有存在4個腔室，針對腔室點錯誤的情況處理**

**parameter:**

CurrentCenters: 輸入一幀的腔室中心點，dict

**return:**

CurrentCenters: 經異常處理後的4個腔室座標，dict

* **Module Name: MuscleSampling.py**

此模組將MultiThreshold 白色區塊(肌肉區域)的輪廓取樣。使用Python Class創建，Class名稱為ConnectBound，以下為ConnectBound必要的輸入參數及屬性。另外，此模組被使用在**A4CGLS**的**MuscleMatching**底下。

|  |  |  |
| --- | --- | --- |
| 參數名稱 | 資料類型 | 說明 |
| src | numpy.ndarray | 輸入影像，3通道或灰階皆可 |
| mask | numpy.ndarray | 要計算的ROI區域mask，二值圖 |
| LevelThres | int | MultiThreshold 的階數門檻，默認為1 |

|  |  |  |
| --- | --- | --- |
| 屬性名稱 | 資料類型 | 說明 |
| self.src | numpy.ndarray | 存放影像 |
| self.mask | numpy.ndarray | 存放ROI的mask |
| self.level | int | 存放LevelThres的值 |
| self.handle | numpy.ndarray | 存放待處理的影像 |
| self.speckle | numpy.ndarray | 將Sampling的點繪製成圖像 |
| self.SamplingInfo | dict | 存放Sampling的點座標資訊 |
| self.\_HandleMTsrc(LevelThres) | method | 過濾圖像中門檻值以下的部分 |
| self.\_Sampling(Cnt, Pos, step) | method | 根據step將肌肉的輪廓區域取出取樣點 |

Example:



Vp 為模組的名稱，使用import module as name做法取為Vp

* **Functions:**

1. **ImageContrast(img[, brightness=0[, contrast=0]]): 調整影像對比度及亮度**

**parameters:**

img: 原始影像，numpy.ndarray

brightness: 亮度大小，大於0則越亮否則反之，默認0，int

contrast: 對比度大小，大於0對比越強烈否則反之，默認0，int

**return:**

img: 調整過後的影像，numpy.ndarray

1. **SplitContour(Cnts): 拆分輪廓(將cv2的輪廓拆分成每個點座標列表)**

**parameter:**

Cnts: cv2的輪廓資訊

**return:**

CntList: 拆分後的結果，np.array

* **Method:**

1. **ContourSampling(self[, step=8]): 將肌肉輪廓取樣**

**parameter:**

step: 每次取樣的間隔，int

**return:** None

Example:



使用object.SamplingInfo 可以取得 Sampling 的座標資訊

1. **\_HandleMTsrc([LevelThres=1]):** **過濾圖像中門檻值以下的部分**

**parameter:**

LevelThres: MultiThreshold 的階數門檻，默認為1，int

**return:**

img: 過濾後的影像(被存放在self.handle 裡面)，numpy.ndarray

1. **\_Sampling(Cnt, Pos, step): 根據step將肌肉的輪廓區域取出取樣點**

**parameters:**

Cnt: 當前的肌肉輪廓，list(contours)

Pos: 輪廓的索引位置，int

step: 取樣間隔，int

**return:** None

* **Module Name: A4CGLS.py**

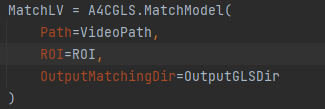
此模組將LV的肌肉區塊進行語意分析，使用Python Class創建，Class名稱為MatchModel，以下為MatchModel必要的輸入參數及屬性。

(註: 由於只處理到matching完後muscle semantic 的部分，因此要新增東西或者往後設計時回再調整名稱)

|  |  |  |
| --- | --- | --- |
| 參數名稱 | 資料類型 | 說明 |
| Path | str | 輸入AVI影片路徑 |
| ROI | list | 使用**Preprocessing的ROI function回傳值當成輸入**，否則會出bug |
| OutputMatchingDir | str | 輸出GLS影片檔案資料夾路徑 |

|  |  |  |
| --- | --- | --- |
| 屬性名稱 | 資料類型 | 說明 |
| self.video | cv2.VideoCapture | 用來讀取影片的屬性 |
| self.Path | str | 存放影片路徑 |
| self.Height | int | 影片高度 |
| self.Width | int | 影片寬度 |
| self.OutputMatchingDir | str | 輸出Semantic影片的資料夾路徑 |
| self.roi | numpy.ndarray | 超音波影像(ROI)區域 |
| self.ox | int | ROI 扇形區域的圓心x座標 |
| self.oy | int | ROI 扇形區域的圓心y座標 |
| self.MuscleSemantic | dict | 紀錄肌肉區段的Sample Points |
| self.AfterBlur | dict | 經過中值濾波的肌肉區段Sample Points |
| self.LeftMValve | list | 左側二尖瓣支點位置(使用**A4CSegmentation** 的 **LeftPivotList 屬性**即可) |
| self.RightMValve | list | 右側二尖瓣支點位置(使用**A4CSegmentation** 的 **RightPivotList 屬性**即可) |

Example:



* **Functions:**

1. **A4CModel(width, height): 處理A4C的標準模型**

**parameters:**

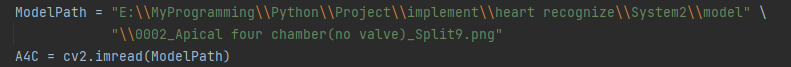
width: 調整模型比例的寬度，int

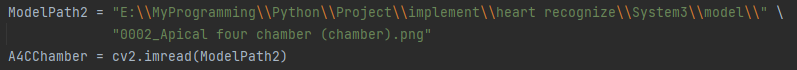
height: 調整模型比例的長度，int

**return:**

A4CModelInfo: A4C模型的資訊，dict。字典包含了Split\_9\_Model、UpperModel、LowerModel、ChamberModel和LVModel的Centers、Contours、Position、src資訊

註: 這裡使用時，需要調整模型的路徑





ModelPath1、ModelPath2 的路徑必須調整成個人電腦裡存放模型的路徑，否則會出錯。

1. **ModelMatching(src, target, vertical, horizontal, theta): 匹配標準模型，並且回傳最佳結果資訊**

**parameters:**

src: Multi-Threshold 的灰階圖像，numpy.ndarray

target: 灰階標準模型，numpy.ndarray

vertical: 垂直參數range(start, end, step)，iterator

horizontal: 水平參數range(start, end, step)，iterator

theta: 角度參數range(start, end, step)，iterator

**return:**

modelBest: 最佳模型擬合結果，灰階圖(模型)，numpy.ndarray

BestFitting: 最佳影像擬合結果，灰階圖(影像)，numpy.ndarray

(BestVertical, BestHorizontal, BestAngle): 最佳擬合垂直、水平、角度參數

* **Method:**

1. **MuscleMatching(self, LeftMValvePos, RightMValvePos[, isOutputVideo=True]): Matching LV肌肉模型後，將LV上的肌肉進行語意分析**

**parameters:**

LeftMValvePos: 左側二尖瓣支點的座標列表

RightMValvePos: 右側二尖瓣支點的座標列表

isOutputVideo: 是否要輸出GLS的影片，默認True，bool

**return:** None

Example:



1. **\_MedBlur(Left, Right): 針對左右側瓣膜支點位置中值濾波**

**parameters:**

Left: 左側二尖瓣支點的座標列表

Right: 右側二尖瓣支點的座標列表