

ModelManager.py

ModelManager 主程式

readDeviceList()

讀取裝置目錄

devicesRequest(deviceDict, devicesReq, modelName)

接收用戶指定的裝置選項

connectDevice(host, port, modelName)

連線至裝置

connectDatabase()

連線至資料庫

registerModel(modelName, uploader, devicesReq)

將 Model 註冊至資料庫中

ModelDatabase

Model 資料庫 lib

connect(dbName)

連接 sqlite 資料庫 dbName

create()

建立 table

getSession()

獲取 Session

ModelFileDB

儲存各 Model 檔 (以及產生 Model 的 python 檔)

FL

CAV Device1--> client1.py(启动联邦学习方式)

DAN.py 與 IoTtalk 連線

CAV Device2--> client2.py(启动联邦学习方式)

name_parameter() 加载模型参数

CAV Device3--> client3.py(启动联邦学习方式)

dataset.py 处理 data 资料集

FML

CAV Device1--> client1.py(启动联邦元学习方式)

DAN.py 與 IoTtalk 連線

CAV Device2--> client2.py(启动联邦元学习方式)

name_parameter() 加载模型参数

CAV Device3--> client3.py(启动联邦元学习方式)

dataset.py 处理 data 资料集

FMLtalk Device

DAN.py

與 IoTtalk 連線

`device_registration_with_rety(URL=None, addr=None)`

註冊裝置

`pull(FEATURE_NAME)`

接收數據

`push(FEATURE_NAME, *data)`

上傳數據

DataCollector.py

收集訓練用數據

`setFilename(FN)`

設定輸出檔案名

`setDataTitle(t)`

設定欄位標題

`dataCollect(l)`

收集數據

`xListToPandas(xl)`

將數據以 Pandas 格式回傳

`printData()`

將數據印在畫面上

`writeFile()`

將數據輸出成 csv 檔案

ModelTrainer.py

訓練 Model

importData(fileName)

導入訓練用數據

modelTrainingWithPytorch(modelName)

使用 Pytorch 的 loss.backward()進行訓練

modelScoringWithPytorch(test_data, test_label, modelName)

使用 Pytorch 的 nn.BCEWithLogitsLoss 套件進行評估

DetectionModule.py

運行檢測

importData(fileName)

導入檢測用數據

detectionWithKeras(data_x, data_y, modelName, output = False, printOnScreen = False)

運行 Model

Model Platform Program

randomAddr()

隨機產生註冊用位址

setSpeedLimit(n)

設定速限

registerDevice()

註冊裝置至 IoTtalk

`ruleBasedDetection()`

以 Rule Based 方式檢測速限

`alert(a)`

回傳速度警告至 IoTtalk

`getResult()`

獲得速限檢測結果

`handler.py`

檢測數據合理性

`randomAddr()`

隨機產生註冊用位址

`setErrorLimit(n)`

設定警告觸發值

`registerDevice()`

註冊裝置至 IoTtalk

`modelTraining(filename, modelName)`

訓練 Model

`throttle_speed_error(throttle, old_speed, new_speed)`

檢測油門與速度關係值

`brake_speed_error(brake, old_speed, new_speed)`

檢測煞車與速度關係值

`speed_gps_error(speed, old_gps, new_gps)`

檢測速度與 Gps 關係值

`ruleBasedDetection(ts_check = True, bs_check = True, sg_check = True)`

以 Rule Based 方式檢測數據關係

`pushValidity()`

回傳檢測結果至 `IoTalk`

`getErrorCount()`

獲得檢測結果

Carla

`dataprocessing(Vehicle_data)`

处理行车数据

`find_weather_presets():`

找到 `CARLA` 中的天气预设

`get_actor_display_name(actor, truncate=250):`

`CARLA` 中一个 `actor` 的显示名称

`on_world_tick()`

在每一帧更新时从世界获取必要的信息

`toggle_info()`

切换显示和隐藏 `HUD` 信息

`render()`

将 `HUD` 内容渲染到屏幕上

DeviceList.csv

`ModelPlatformd` 的 `Ip/port` (`Manager` 會自動讀取此檔)

template

Model Platform and Manager GUI 頁面

index.html

首頁

upload.html

上傳頁面

success.html

上傳成功頁面(自動跳轉)

leaderboard_sedlect.html

排行榜項目選擇頁面

leaderboard.html

排行榜頁面

bootstrap

csv/js 模板