# 環境建置

# **影像辨識**

## 安裝軟體

1. NVIDIA顯示卡驅動411.31以上
2. Visual Studio Community 2017
3. CUDA Toolkit 10.0
4. NVIDIA cuDNN 7.6.5.32(對應CUDA Toolkit 10.0)
5. Anaconda 3
6. OpenCV 3.4.11
7. Darknet()

## 編譯Darknet

1. 進入控制台/系統及安全性/進階系統設定/進階/環境變數  
   尋找「系統變數System」中的「Path」，新增以下變數：  
   C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin  
   C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\lib\x64  
   C:\Program Files (x86)\Microsoft Visual Studio  
   \2017\Community\VC\Tools\MSVC\14.15.26726\bin   
   (VS 根據版本路徑會有不一樣，請找到VC下的bin資料夾為原則)
2. 使用Visual Studio 開啟darknet\build\darknet\darknet.sln
3. 在專案darknet上按右鍵，選擇property(屬性)。
4. 將Configuration(組態管理員)選擇Release，Platform(平台)選擇x64
5. VC++ Directories → Include Directories → 依據opencv安裝位置設定路徑  
   須設定  
   opencv\build\include   
   opencv\build\include\opencv  
   opencv\build\include\opencv2
6. VC++ Directories → Library Directories → 依據opencv安裝位置設定路徑  
   opencv\build\x64\vc14\lib
7. Linker → Input → Additional Dependencies → 加入opencv\_world3411.lib  
   opencv\_world3411.lib  
   (根據opencv版本不同，檔名會有所不同，根據opencv\build\x64\vc14\lib底下檔案名稱為準)
8. 將cudnn的三個資料夾bin、include、lib中的檔案(共三個檔案)  
   複製到cuda的安裝目錄：C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0 相應的資料夾內
9. CUDA C/C++ → CUDA Toolkit Custom Dir → C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0
10. CUDA C/C++ → Device → Code Generation 內參數全部移除
11. 把opencv\build\x64\vc14\bin中的3個.dll檔案複製至C:\Windows\System32底下
12. 按下Visual Studio 2015 的Debugger 進行編譯  
    如果編譯成功，就會在x64資料夾下多一個darknet.exe檔案
13. 使用Visual Studio 開啟yolo\_cpp\_dll.sln
14. 重複執行3.~12.步驟\
15. 開啟Terminal 進入x64資料夾，就可以測試簡單範例  
    .\darknet.exe detect .\cfg\yolov4.cfg .\weights\yolov4.weights .\data\dog.jpg
16. 將x64中的yolo\_cpp\_dll.dll與pthreadVC2.dll複製至nutc-aiot-server/sign\_detect/ObjectDetection/
17. 將yolov4.weights複製至nutc-aiot-server/sign\_detect/ObjectDetection/
18. 執行main.py